



SLSTR: Algorithm Theoretical Basis Definition Document for Level 1 Observables

Prepared by: Andrew Birks/Caroline Cox Date _____

Checked By: Dave Smith Date _____

Project Manager: Dave Smith Date _____

PA Approval: Dion Dawson Date _____



Distribution List

RAL	A.R. Birks C. Cox T.J. Nightingale H. Mortimer A.R. Birks C.T. Mutlow D. Dawson B.J. Maddison
Thales Alenia	D. Lebedef J. L. Palmade
ESA	J. Nieke
ACRI	L. Bourg I. Muguret



Change Record

Issue	Date	Changes
Draft A	13/10/06	Original Version (Outline only)
1.0	20/12/06	New Revision
1.1	05/03/08	Overview revised
1.2	18/06/08	Revised draft: Section 2.4.5 outlining the theory of the geolocation added. Geolocation equations revised to respect SLSTR scan geometry in Section 3.2. Additional material added to cloud clearing sections. Page and equation Numbering revised; various minor corrections
1.3	17/07/08	Revised; Sections on regridding, cosmetic fill and conversion to radiance added. Various minor corrections.
1.4	16/12/08	1.6 micron cloud test added. Placeholders added for tests based on the new 1.375 and 2.25 micron SLSTR channels.
1.5	13/03/09	Various comments on v1.4 incorporated Document format changed to reflect common project style Sections renumbered Equations Renumbered L0 Processing Section included in section 5.1 Scope of TDI processing described in section 5.3.3.10 Conversion from reflectance to radiance added 5.5.3.8
1.6	12/06/09	Housekeeping packet unpacking added. Infrared calibration updated. Infrared histogram cloud test details added and new 1.375 micron cloud test defined.
1.7	15/12/09	TDI Section removed since this is not performed at L1. Instead simple average of a and b pixels will be performed. Stage 18 of L1b processing updated accordingly. (Section 5.2.2) Updated reference RD-2, RD-3 and RD-13 (Section 2.2) Acronym List Updated (Section 3.2) Expanded description of L1b geolocation (Section 4.4.4) Scan jitter test updated for SLSTR (Section 5.4.3.1.7.2) BBEU and TAEO conversion algorithms updated although these are still pending information from the supplier. (Section 5.4.3.1.7) Description of science data (L0) processing (Section 5.4.3.2) Surface classification added (Section 5.5.3.6) Various Claifications and Typographical Corrections (All) Breakpoint Table Added (Appendix A)



1.8	08-July-2010	<p>Update in response to numerous comments from ACRI, TAS-F and ESA.</p> <p>Consistent terms in calibration equations. Offset and slope used for calibration equations. Other edits made to calibration section</p> <p>Added sub-sections on Meteo annotation fields to the relevant sections (5.5..5 L1b processing, 5.5.2 processing objective and 5.5.3 mathematical description) based on OLCI descriptions.</p> <p>Scene classification reverted to simple 'centre of pixel' scheme.</p> <p>Solar channel calibration clarifications added</p> <p>Nomenclature of conversion from reflectance to radiance modified for consistency with previous sections.</p> <p>Purge of AATSR processing internal variables. Exception values defined explicitly.</p> <p>Figure numbers in section 4 corrected</p> <p>Reordering of Ancillary conversion functions to be compliant with SY-4</p>
1.9	13/07/2010	<p>Section 5.4.3.1.6 - Additional ancillary conversion function 'f20: Look-Up Table' inserted on request from TAS-F.</p>
2.0	14/01/2011	<p>Changes implemented from S3-MO-RAL-SL-2010-01 - Issue 9 - Change Sheet for SLSTR ATBD.</p>



Contents

1	INTRODUCTION	7
1.1	Overview	7
1.2	Document Context and Scope	7
2	Documents	8
2.1	Applicable Documents	8
2.2	Reference Documents	8
3	Terms, Definitions and Abbreviations	10
3.1	Terms and Definitions	10
3.2	Glossary of Acronyms	11
4	OVERVIEW AND BACKGROUND INFORMATION.....	13
4.1	Objectives of SLSTR Level 0 and Level 1 Processing	13
4.1.1	Level 0 Processing	13
4.1.2	Level 1 Processing	13
4.2	Instrument Overview	13
4.2.1	The SLSTR instrument.....	13
4.2.2	The measurement cycle.....	14
4.2.3	Terminology and indexing	15
4.3	Calibration Approach.....	16
4.3.1	On-board calibration.....	16
4.3.2	Vicarious Calibration and Validation	16
4.4	Level 1 Products Definition	16
4.4.1	Level 1a.....	18
4.4.2	Level 1b.....	18
4.4.3	The Level 1b product grid	18
4.4.3.1	Calculation of Latitude and Longitude of grid (or image) Pixels	19
4.4.4	Geo-referencing instrument samples	22
4.4.4.1	Frames of reference.....	22
4.4.4.2	The yaw steering frame.....	22
4.4.4.3	The spacecraft reference frame	22
4.4.4.4	The platform reference frame.....	23
4.4.4.5	The instrument reference frame.....	23
4.4.4.6	The scan reference frames	23
4.4.4.7	Satellite transformations.....	23
4.4.4.8	Treatment of multiple detectors.....	26
4.4.4.9	Geometry on the Ellipsoid	28
4.4.4.10	Calculation of Grid coordinates of Scan Pixel	38
5	ALGORITHM DESCRIPTION.....	43
5.1	Level 0 Processing for the SLSTR Instrument.....	43
5.1.1	The SLSTR Instrument Science Packet	43
5.1.1.1	Comments	47
5.1.2	The SLSTR Level 0 ISP Product.....	48
5.1.2.1	ISP File Unpacking.....	48
5.1.2.2	ISP Index File	51
5.1.3	NAVATT File Processing	54
5.2	Overview of Level 1B processing for the SLSTR Instrument.....	55
5.2.1	Level 1a Processing	56
5.2.2	Level 1b Processing	59
5.3	Geo-referencing	62
5.3.1	Algorithm Input	62
5.3.2	Processing Objective	62
5.3.2.1	Satellite Time Calibration (Stage 9)	63
5.3.2.2	Geolocation (Stage 10-14)	63
5.3.3	Mathematical Description	64



5.3.3.1	Notation	64
5.3.3.2	Common Procedures	65
5.3.3.3	Satellite Time Calibration	68
5.3.3.4	Generate Geolocation Grid	70
5.3.3.5	Satellite Navigation and Attitude	74
5.3.3.6	Geolocate Pixels	74
5.3.3.7	Calculate Pixel x-y co-ordinates	81
5.4	Level 1a Processing	86
5.4.1	Algorithm Input	86
5.4.1.1	Level 0 Instrument Digital Numbers	86
5.4.1.2	Ancillary Radiometric Data	86
5.4.2	Processing Objective	86
5.4.2.1	Source Packet Processing (Stages 1 - 6)	86
5.4.2.2	Infra-Red Channel calibration (Stage 7)	86
5.4.2.3	Visible Channel calibration (Stage 8)	87
5.4.3	Mathematical Description	88
5.4.3.1	Source Packet Processing	88
5.4.3.2	Science Data Processing (Stage 6)	101
5.4.3.3	Infra-red channel calibration	103
5.4.3.4	Solar channel calibration	107
5.5	Level 1b Processing	115
5.5.1	Algorithm Input	115
5.5.1.1	Level 1a Data	115
5.5.1.2	Auxiliary Data	115
5.5.2	Processing Objective	115
5.5.2.1	Signal Calibration (Stage 17)	115
5.5.2.2	Time Domain Integration (Stage 18)	115
5.5.2.3	Regrid Pixels (Stage 19)	116
5.5.2.4	Cosmetic Fill (Stage 20)	116
5.5.2.5	Image Pixel Positions (Stage 21)	116
5.5.2.6	Determine Land-Sea Flag (Stage 22)	116
5.5.2.7	Cloud clearing (Stage 23)	116
5.5.2.8	Meteo annotations (stage 24)	116
5.5.3	Mathematical Description	117
5.5.3.1	Calibration of Pixel Data	117
5.5.3.2	Time Domain Averaging of SWIR Channels	120
5.5.3.3	Regrid pixels	123
5.5.3.4	Cosmetic Fill	132
5.5.3.5	Surface classification	136
5.5.3.6	Identification of Cloud Affected Pixels	137
5.5.3.7	Convert Solar Channel Reflectances to Radiance	170
5.5.3.8	Meteo annotations	171
Appendix A – Breakpoint Table		172



1 INTRODUCTION

1.1 OVERVIEW

The main scope of the S3 mission is to serve research relating to the marine environment, with the main objective being to determine such parameters as sea surface topography, sea and land surface temperature and ocean colour. However, the mission will also support various land, atmospheric and cryospheric applications. The first S3 satellite is expected to launch in 2013, followed by a second launch in order to provide maximum coverage.

The Sea and Land Surface Temperature Radiometer (SLSTR) forms part of the S3 payload and is an infrared and visible radiometer for ocean and land monitoring. The instrument is described in more detail in section 4.2.1.

1.2 DOCUMENT CONTEXT AND SCOPE

This Algorithm Theoretical Basis document (ATBD) describes the algorithms used to produce the SLSTR Level 1 Radiometric Product and accompanying geo-location and environment data. In particular the document is concerned with the algorithms to be implemented in the SLSTR O-GPP.



2 DOCUMENTS

2.1 APPLICABLE DOCUMENTS

AD-1	Deleted reference – no longer applicable		
AD-2	Deleted reference – no longer applicable		
AD-3	GMES Sentinel-3 System Requirements Document	S3-RS-ESA-SY-0010	Issue 4.0, 13-Nov-09
AD-4	Earth Explorer Mission Software. EXPLORER_POINTING Software User Manual	EE-MA-DMS-GS-0005	Issue 4.1, 07-May-10

2.2 REFERENCE DOCUMENTS

RD-1	Contributions to Ground Segment Service Segment Requirements	SENS3-RAL-TN-31	24-Aug-2006
RD-2	SLSTR Design Description	IN-2 S3-RP-GA-SL-00008	07-Nov-2008
RD-3	SLSTR L1 Products Calibration and Validation Plan	S3-PL-RAL-SL-008	18-Jul-2008
RD-4	Sentinel-3 S3 Geometrical Performance Budgets Part 1 : General rules and contributors	SY-6 c, Issue 1, S3-TN-TAF-SY-01874	20-Dec-2010
RD-5	SLSTR: Science data Software Interface Requirement Document	S3-IS-GA-SL-00003	11-Nov-2008
RD-6	SLSTR Instrument Measurement Data Definition Document	S3-RP-GA-SL-00019	Dec-2009
RD-7	Zavody, A.M., Mutlow, C.T., and Llewellyn-Jones, D.T. Cloud Clearing over the Ocean in the Processing of Data from the Along-Track Scanning Radiometer (ATSR). J. Atmos. Oceanic Technol., <u>17</u> , 595 – 615, 2000.		
RD-8	Space Packet Protocol	CCSDS 133.0-B-1	
RD-9	Deleted reference – no longer applicable		
RD-10	GPP Perimeter and Mission Products Format	S3-MO-TAF-00424/2008	
RD-11	SLSTR L1 Products Definition	S3-RS-RAL-SY-003	Issue 5, 14-Jan-2011



RD-12	Discriminating clear-sky from cloud with MODIS:Algorithm Theoretical Basis Document (MOD35)	MODIS Cloud Mask Team (Ackerman, s. et al)	Team	Oct-2006
RD-13	Geodesy (4 th Edition)	G. Bomford	Clarendon Press, Oxford	
RD-14	NAVATT Packets Definition	S3-MO-TAF-00886/2009		04-Nov-2009



3 TERMS, DEFINITIONS AND ABBREVIATIONS

3.1 TERMS AND DEFINITIONS

Accuracy is defined as the difference between a result obtained and the 'true' value.

Channel- Spectral channel (S1-S9 + F1-F2)

Chain - Analogue channel of the signal processing chain

Detector - Pixel array at band N

Calibration is the process of quantitatively defining the system response to known, controlled system inputs.

Field-Of-View - is the angular extent of a given scene that is viewed by the instrument.

Infrared (IR) radiation is electromagnetic radiation of wavelengths between about 750nm and 1mm. This is broken down into 5 wavelength regions

Near-IR - 0.75-1.4 μm

Short-Wave IR - 1.4-3 μm

Medium-Wave IR - 3-8 μm

Long-Wave IR - 8-15 μm

Image swath - Maximum distance on ground between the positions of two spatial samples belonging to the same row.

Image sample - image element containing radiance measurements of co-registered pixels for all bands

Pixel - Detector single element, considering also fire elements (s7), blind and TDI ones

Detector 5 has 4x2 pixels (S5) + 2 reference pixels (blind) ,

Detector 7 has 2x1 pixels (S7) + 1 reference blind + 4x1 fire pixels (F1)

Precision is defined as the difference between one result and the mean of several results obtained by the same method, i.e. reproducibility (includes random errors only).

Traceability is the property of the result of a measurement or the value of a standard whereby it can be related to stated references, usually national or international standards, through an unbroken chain of comparisons all having stated uncertainties.

Visible light is electromagnetic radiation detectable by the human eye with a wavelength between approximately 400nm and 700nm.

Validation is the process of assessing by independent means the quality of the data products derived from the system outputs.



3.2 GLOSSARY OF ACRONYMS

(A)ATSR	(Advanced) Along Track Scanning Radiometer
ADS	Annotation Data Set
AST	Averaged Surface Temperature (AATSR product)
ATSR-2	Along Track Scanning Radiometer-2
ATBD	Algorithm Theoretical Basis Document
BB	Black Body
BBEU	Black Body Electronics Unit
CCSPS	Consultative Committee for Space Data Systems
CFI	Customer Furnished Item
CPE	Control and Processor Electronics (component of the SLSTR)
DDT	Data Dictionary Tool
Defra	UK Department for Environment, Food and Rural Affairs
DEM	Digital Elevation Model
DPM	Detailed Processing Model
DS	Data Set
DSD	Data Set Descriptor
DSR	Data Set Record
ESA	European Space Agency
ESL	Expert Support Laboratory
FAT	Factory Acceptance Test
FEE	Front End Electronics (component of the SLSTR)
FEP	Front End Processor
FPA	Focal Plane Assembly
FOS	Flight Operations Segment
GBTR	Gridded Brightness Temperature / Reflectance (AATSR product)
GMES	Global Monitoring for Environment and Security
GST	Gridded Surface Temperature (AATSR product)
IFOV	Instantaneous Field of View
IODD	Input-Output Data Definition
IPF	Instrument Processing Facility
ISO	International Organisation for Standardisation
ISP	Instrument Source Packets
L1b	Level 1b
L2	Level 2
LBR	Low Bit Rate
LST	Land Surface Temperature
LUT	Look Up Table
LWIR	Long Wave InfraRed
MDS	Measurement Data Set
MDSR	Measurement Data Set Record
MERIS	MEdium Resolution Imaging Spectrometer
MPH	Main Product Header
MTF	Modulation Transfer Function
MWIR	Medium Wave InfraRed
NRT	Near Real Time
OP	Operational Processor
O-GPP	Optical Ground Prototype Processor
OLCI	Ocean and Land Colour Instrument
PADD	Product Algorithm Detailed Documentation
PCAT	?
PCD	Product Confidence Data



PDS	Payload Data Segment
PP	Prototype Processor
PUS	Packet Utilisation Standard
QC	Quality Control
QWG	Quality Working Group
RFQ	Request for Quotation
RT	Radiative Transfer
SAFE	Standard Archive Format for Europe
SLSTR	Sea and Land Surface Temperature Radiometer
SOW	Statement of Work
SPH	Specific Product Header
SPS	System Performance Simulator
SST	Sea Surface Temperature
SUM	Software User Manual
SWIR	Short Wave InfraRed
TAEU	Temperature Acquisition Electronics Unit
TBC	To be confirmed
TD&P	Test Definition and Procedures
TDI	Time Domain Integration
TDS	Test Data Set[s]
USF	User Service Facility
WP	Work package
XML	eXtensible Markup Language



4 OVERVIEW AND BACKGROUND INFORMATION

4.1 OBJECTIVES OF SLSTR LEVEL 0 AND LEVEL 1 PROCESSING

4.1.1 Level 0 Processing

The SLSTR Level 0 product will consist of a SAFE container structure containing NAVATT and SLSTR data sets from the satellite telemetry, these data sets comprising Header, Annotation, and Data files. In particular the SLSTR data files will be binary files containing SLSTR Instrument Source Packets. The input to the Operational Processor will consist of the Level 0 product from the FEP. In the context of the OP, therefore, Level 0 processing will consist of unpacking the ISP and annotation data into suitable data structures from which it is available for onward processing.

It is not intended that the Optical SPS shall include any emulation of the FEP. Instead the O-GPP will receive simulated ISP and NAVATT packets from the SPS, stored in separate files. The principal objective of the O-GPP is to support the validation of the Operational Processor, and so in the case of the GPP Level 0 processing will be required to read the binary ISP files, unpack them in to the data structure as above, and assemble them into the Header, Annotation, and Data files of the Level 0 product proper for output.. At the same time, in order to allow two-way transport between the OP and GPP, the functionality to input and unpack the data as in the previous paragraph will be required.

It has been agreed that for transport between the O-GPP and the OP, it will not be a requirement that the component level 0 files will be included in the SAFE container (which is primarily required for archiving purposes).

4.1.2 Level 1 Processing

The objective of Level 1 processing is to derive, starting from the time-ordered and consolidated instrument source packets, a data product containing calibrated, geolocated brightness temperatures, for the thermal channels, and radiances for the short wave and visible channels, together with appropriate annotation data. These form the starting point of the Level 2 algorithms for retrieval of geophysical parameters such as SST.

4.2 INSTRUMENT OVERVIEW

4.2.1 The SLSTR instrument

A principal aim of the Sentinel SLSTR instrument mission is to maintain continuity with the (A)ATSR series of instruments, and the proposed design supports this by incorporating the basic functionality of AATSR, with the addition of some new, more advanced, features. These include a wider swath, new channels, and higher spatial resolution in some channels.

The proposed instrument will include the set of channels used by ATSR-2 and AATSR, thus ensuring continuity of data, together with two new channels at wavelengths of 2.25 and 1.375 microns in support of cloud clearing for SST retrieval. In total the instrument will have eleven channels as follows:

- 3 thermal infra-red channels at 3.7, 10.8 and 12 micron wavelengths;
- 2 fire channels at 3.7, and 10.8 micron wavelengths;
- 6 short-wave and visible channels at 2.25, 1.6, 1.375, 0.87, 0.67 and 0.55 micron wavelengths.

Like AATSR, the Sentinel SLSTR instrument will measure a nadir and an oblique view scan, each of which will also intersect the calibration black bodies and the visible calibration unit. The requirement for a wider nadir swath, and for enhanced resolution in coastal regions, means that the number of instrument



pixels to be measured per scan will exceed the corresponding figure for AATSR. The baseline design, presented at the PRR, accomplishes this while reducing the scan rate.

AATSR scans at a rate of 400 scans per minute, corresponding to the along track resolution of 1 km. The SLSTR instrument will use two independent scan mirrors each scanning at 200 scans per minute, but each scan will measure 2 along-track pixels of 1 km (and 8 pixels at 500 m resolution) simultaneously. This configuration increases the swath width in both views, as well as providing 500 metre resolution in the reflectance channels.

Each scan mirror is mounted at an oblique angle to its axis of rotation, and directs radiation into a telescope assembly the optical axis of which is aligned parallel to the rotation axis. As the scan mirror rotates, the line of sight traces out a cone whose intersection with the Earth traces out the measurement swath of the instrument. The scan cone will intersect the Earth view, the two calibration black bodies, and the Visible Calibration (VISCAL) Unit, so that the line of sight will encounter each of these once during a complete rotation.

Radiation incident along the line of sight enters the focal plane assembly, where it is split into frequency bands corresponding to the different channels. Radiation in each channel is focussed onto a small array of detector elements which correspond to pixels.

4.2.2 The measurement cycle

The two scan mechanisms rotate in synchronism at a frequency of 200 cycles per minute. The basic timing relationships are described in reference [RD-6], and are as follows. The scan cycle has a period of 300 ms (200 rotations per minute). There is also a readout cycle or acquisition sequence of 2 scan periods (600 ms), because the black body and VISCAL targets for a given view are only sampled on alternate scans. These timing cycles are identified in reference [RD-6] as SCANSYNC and CYCLESYNC respectively.

Within these envelopes we have two sampling frequencies which govern the sampling of the individual detectors. In the terminology of [RD-6] these are:

- PIX10SYNC: There will be exactly 3670 (TBC) PIX10SYNC periods in the 300 ms SCANSYNC period, so its duration is 81.7 μ s. This is the rate at which the thermal (1 km resolution) channels are sampled.
- PIX05SYNC: This corresponds to a period which is exactly one-half of PIX10SYNC, or 40.9 μ s. This defines the frequency at which the reflectance channels, having 0.5 km ground resolution, are sampled.

As the acquisition cycle proceeds the detectors are sampled simultaneously and synchronously at these frequencies. The thermal infra-red (LWIR and MWIR) and fire channels are sampled once per PIX10SYNC interval. Since each detector in these channels has two detector elements, this process gives two samples per PIX10SYNC period in each channel.

The SWIR and visible channels are sampled twice per PIX10SYNC interval, on alternate PIX05SYNC pulses, to give a ground resolution of 0.5 km. In the tables of ISP structure in reference [RD-5] the odd and even PIX05SYNC intervals appear to be identified as 'cycle 1' and 'cycle 2' respectively. In the case of the SWIR channels each detector in the FPA will have 8 detectors arranged in an array of two columns by four rows (with the long side of the array parallel to the ground track), to give 0.5 km sampling in the along-track direction.

Not all of the samples within a scan cycle appear in the telemetry, since there is no point in including samples taken when the line of site is viewing the inside of the instrument enclosure. Only samples corresponding to the earth view or the calibration targets are transmitted to the ground. As previously noted, the black body and VISCAL targets for a given view are only sampled on alternate scans. Figure



4-1, derived from the description in [RD-6], indicates the target measurements that are recorded during each scan cycle, for the nadir and inclined views.

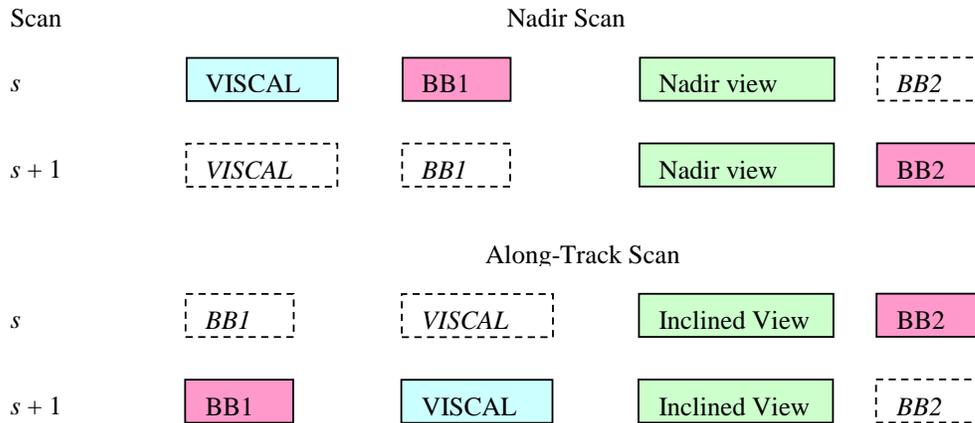


Figure 4-1: The SLSTR Measurement Cycle. Dashed boxes represent target encounters that are absent from the telemetry

4.2.3 Terminology and indexing

We define the following terminology

- A **scan** is defined as a complete rotation of the scan mirrors.
- A **scan cycle** comprises two consecutive scans (a CYCLESYNC period as defined above) during which a complete set of targets are sampled.
- A **target** is either the Earth view or one of the internal calibration targets (the VISCAL and the two black bodies); in the context of the telemetry data it refers to a section of the scan during which valid data is obtained, when the detectors are viewing the earth view or one of the calibration targets.
- A **pixel** is the IFOV of a single detector element; the projection of that detector element onto the ground at a given instant.
- A **scan trace**, or **scan locus** is the trace of a single detector element on the ground. Thus for example in the thermal channels each detector has two elements, and so a single scan will give two scan traces, displaced by 1 km in the along-track direction. Adjacent scan traces represent adjacent 'rows' of the instrument grid. We may use the term **instrument scan**, (as distinct from *scan*), as an alternative to *scan trace*.

During operation, the SLSTR Front End Electronics will transmit a packet of data to the CPE every PIX10SYNC interval, containing all the detector samples recorded during that interval, and these groupings are retained in the ISP transmitted to ground. This means that to uniquely identify a pixel in a given channel we may need the following indices:

- S Scan number; a continuous count of scans, derived from the scan counter that we assume will be present in the telemetry, but with overflow corrected, and with an origin close to the start of the product.



- p Pixel index; an index to the PIX10SYNC interval within which the sample was measured.
- t An index to the PIX05SYNC interval within which the sample was measured; $t = 0, 1$ in the case of the 0.5 km (reflectance) channels,
- k Detector index. $k = 0, 1$ for the 1 km channels, $k = 0, \dots, 7$ for the 0.5 km channels. The detector index maps on to a two dimensional matrix of detectors, the row and column of which are aligned in the along-track and across-track directions respectively, and so the mapping between k and the two-dimensional row and column indices needs to be defined.
- s Scan trace; for the thermal channels, $s = 2S + \text{row}(k)$, for the reflectance channels, for which the detector arrays have 4 rows, $s = 4S + \text{row}(k)$

4.3 CALIBRATION APPROACH

4.3.1 On-board calibration

On-board calibration of the Sentinel SLSTR will be based on the approach successfully implemented in the (A)ATSR series of instruments.

For the thermal infra-red channels, provision for their continuous on-board calibration is an integral part of the instrument design. The instrument includes two stable and high precision black-body targets, which are crossed by both the nadir and along-track scan cones. Thus each target is observed once per scan on each scan. One black body is held at a temperature lower than the expected range of surface brightness temperature, and the other at a temperature higher the expected range, so that the interval between the two black body temperatures covers the expected range of sea surface brightness temperature. The physical temperatures of the two black bodies are continuously monitored, and so for each scan the linear calibration relationship between the infra-red radiance and the measured signal counts can be determined from the measurements of the black body signals.

The use of two blackbodies in this way ensures that the calibration is optimised over the normal range of SST, and the effects of detector non-linearity over this range are minimised, thus maximising the accuracy of the SST measurement. The treatment of detector non-linearity is discussed further below.

The calibration scheme for the short-wave (near infra-red) and visible channels is based on a diffuse calibration target of accurately known reflectance which is illuminated by the sun over a short segment of the orbit, and which is intersected by the instrument scans. Measurements of this target when it is illuminated by the sun enable the linear calibration relationship between the measured signal in each channel and the surface reflectance to be determined. If applicable, corrections for any detector non-linearity measured during the pre-launch characterisation of the instrument may be applied based on look-up tables.

4.3.2 Vicarious Calibration and Validation

The concept of vicarious calibration (as distinct from validation) is not applicable to the calibration of the thermal infra-red channels of the SLSTR instrument.

Long-term drifts of the calibration of the visible and near-visible channels may be characterised by monitoring the signals from geophysical targets (e.g. certain desert regions) known to show stable reflectance characteristics, and by comparisons with other instruments. Drifts so determined will be used to update the calibration files for the relevant channels. This, and instrument validation, are discussed in more detail elsewhere (reference [RD-3]).

4.4 LEVEL 1 PRODUCTS DEFINITION

Reference [RD-1] defines a single SLSTR Level 1 product, based on the AATSR Level 1B product. Note however that there is some ambiguity of terminology here, and the designation Level 1B does not have the same meaning in the context of Envisat as has been defined for Sentinel 3. In terms of the latter



definition the product conforms to level 1c, since pixels in the two views are re-sampled and collocated onto a common grid

The product as defined in reference [RD-1] includes 34 measurement data sets, each representing the image for a particular channel and view combination as follows:

- 3 sets of calibrated nadir view brightness temperatures for the 3.74, 10.8 and 12 micron channels;
- 2 sets of calibrated nadir view brightness temperatures for the 3.74, and 10.85 micron fire channels
- 6 sets of calibrated nadir view radiances for the 2.25, 1.6, 1.375 (A-Stripe), 0.87, 0.67 and 0.55 micron channels;
- 3 sets of calibrated nadir view radiances for the 2.25, 1.6, 1.375 micron channels B-stripe detectors
- 3 sets of calibrated nadir view radiances for the 2.25, 1.6, 1.375 micron channels TDI (Average of A and B stripe)
- 3 sets of calibrated oblique view brightness temperatures for the 3.7, 10.8 and 12 micron channels;
- 2 sets of calibrated obliqueview brightness temperatures for the 3.74, and 10.85 micron fire channels
- 6 sets of calibrated obliqueview radiances for the 2.25, 1.6, 1.375 (A-Stripe), 0.87, 0.67 and 0.55 micron channels.
- 3 sets of calibrated oblique view radiances for the 2.25, 1.6, 1.375 micron channels B-stripe detectors
- 3 sets of calibrated oblique view radiances for the 2.25, 1.6, 1.375 micron channels TDI (Average of A and B stripe)

The brightness temperature and radiance values will be mapped onto a rectangular grid of resolution appropriate to the channel; this will be 1 km for the thermal IR channels, and 0.5 km for those channels that are measured at this resolution.

Following (A)ATSR practice, the 1 km grid will be defined with respect to the subsatellite track and the instrument swath projected onto the reference ellipsoid WGS84. The width of the nominal image swath will be 1470 km for the nadir view, and 768 km (TBC) for the along-track view. The question of other possible grids will be discussed further in Section 4.4.4 below.

A further four measurement data sets will define, for each pixel at 1 km resolution, cloud and confidence flags:

- nadir view cloud clearing/land flagging;
- oblique view cloud clearing/land flagging;
- confidence words for nadir view pixels;
- confidence words for along-track view pixels.

Supplementary (annotation) data sets will provide, where appropriate on a low resolution grid of 16 by 16 km (TBC):



- latitude and longitude of tie point pixels;
- Cartesian (x, y) co-ordinates for tie point instrument pixels;
- Solar azimuth and elevation at the tie pixel;
- Satellite azimuth and elevation at the tie pixel;
- Topographic correction (over land)
- Scan and pixel number for each pixel (to retain the relationship between instrument and image pixels).

The Sentinel 3 System Requirements Document [AD-3] defines 3 processing levels for level 1 data, Level 1a, Level 1b and Level 1c. The relationship of these to the SLSTR product processing is discussed in the following sections.

4.4.1 Level 1a

Level 1a data is defined as 'Level 0 data with corresponding radiometric, spectral and geometric (i.e. Earth location) correction and calibration computed but not applied'.

This represents a very low processing level, and has no counterpart in the Envisat AATSR product set on which the definition of the SLSTR Level 1 product [RD-1] has been based. Nevertheless, a requirement for such a product has been identified, for long-term calibration and monitoring.

For the ATSR-1 and ATSR-2 instruments a product, the Ungridded Detector Count (UCOUNTS) Product, was defined that represented this level of processing. The product contained uncalibrated channel counts on the instrument grid, with geolocation data. The AATSR Prototype Processor also produces a UCOUNTS product, but no such product is generated by the AATSR ground segment.

4.4.2 Level 1b

Level 1b data is defined in reference [AD-3] as Level 1b data that is calibrated and geolocated but not resampled. However, it has subsequently been agreed that, in the case of the SLSTR, the Level 1b product will be relocated onto a specified quasi-Cartesian product grid defined in the following section. This is in order to fulfil the user requirement for a product that is consistent with, and shows continuity with, the AATSR heritage. (We prefer the term regridded rather than resampled to describe this, since resampling may imply an interpolation.)

This use of the term 'Level 1b' is similar to Envisat usage, and will be adopted in the following.

For the ATSR-1 and ATSR-2 instruments a product, the Ungridded Brightness Temperature / Reflectance (UBT) Product was defined at the level of processing corresponding to the SRD [AD-3] definition above. The product contained calibrated data on the same instrument grid as the UCOUNTS product defined above, along with geolocation data. The AATSR Prototype Processor also produces a UBT product, but again no such product is generated by the AATSR ground segment.

4.4.3 The Level 1b product grid

For each view (nadir and along-track) the SLSTR data is measured on a curvilinear grid defined by the intersection of the instrument scan with the surface of the earth. Instrument samples are distributed at uniform angular intervals around a scan line that is the intersection of the relevant scan cone with the surface of the earth; these scan lines approximate to sections of an ellipse (if the Earth were flat, they would be arcs of ellipses). The two views are not collocated at this stage.

Level 1 processing calculates the geolocation with respect to this instrument grid. That is to say, for each instrument pixel, the latitude and longitude of the pixel are determined. Following AATSR practice, the X



and Y co-ordinates of the instrument pixel are also calculated with respect to a quasi-Cartesian product grid that is defined with respect to the satellite ground track and instrument swath. These values are then used to resample the data onto the product grid; a nearest neighbour algorithm is used.

The definition of this product grid is as follows. Consider a point P in the SLSTR instrument swath. The X co-ordinate of the point is given by its distance from the instrument ground track, measured along the normal section PQ through P that intersects the satellite ground track at right angles; point Q is the intersection point of the normal curve with the ground track. The Y co-ordinate of P is then the distance of the intersection point Q measured along the ground track from an origin point. For consolidated products this origin is the ascending node of the ground track on the equator; for near real time products an arbitrary origin close to the start of the product will be used.

X and Y are then the co-ordinates of the point P in a quasi-Cartesian system whose Y axis is the local tangent to the ground track; we call it quasi-Cartesian because the ground track is curved (it is not a geodesic on the ellipsoid). This defines the 'specified grid' in the sense of the Level 1b definition above.

Given the Sentinel 3 orbit, the co-ordinates X and Y form a well-defined function of latitude and longitude on the reference ellipsoid (the WGS-84 ellipsoid in the case of Sentinel 3). The level 1b product grid is uniformly sampled in the X and Y co-ordinates.

Image points are sampled at uniform intervals of 1 km or 0.5 km, for the thermal and solar channels respectively, in the across-track (X) direction. In the along-track (Y) direction, following AATSR practice, an equal time interval sampling scheme will be adopted. That is to say, the sampling interval in Y will be equal to the distance moved by the sub-satellite point in $(0.5 \times \text{SCANSYNC})$ intervals, for the thermal channels, or $(0.25 \times \text{SCANSYNC})$ intervals for the solar channels. These distance increments approximately equal 1.0 (0.5) km, but will vary slightly around the orbit as the ground trace velocity changes.

This method of sampling has certain advantages, in particular it minimises the incidence of cosmetic fill pixels in the nadir view, and retains continuity with the AATSR algorithms. It has no adverse impact on Level 1c product generation.

The latitude and longitude corresponding to any given value of X and Y can be calculated provided the ground trace of the sub-satellite point is known. In practice Level 1b processing will compute these quantities at a series of tie points corresponding to the product grid sampling points, to populate the Image Pixel Geolocation ADS. Latitude and longitude of intermediate points can then be retrieved by interpolation. Details of how this grid is constructed are given in Section 4.4.3.1 below.

4.4.3.1 Calculation of Latitude and Longitude of grid (or image) Pixels

In order to permit the geolocation of the image pixels, Level 1b processing will compute a table of latitude and longitude at a series of tie points in X and Y corresponding to product grid sampling points. The table serves two purposes. Firstly, the column of points at $x = 0$ defines the ground track, for use by the geolocation algorithm for instrument pixels.

Secondly, the table is used to populate the Image Pixel Geolocation ADS. The latitude and longitude of any image pixel can then be retrieved by interpolation in the table.

The table is produced by the following steps.

Firstly, an auxiliary table is derived giving the latitudes and longitudes of a series of points on the ground track, spaced in time by a multiple of the SCANSYNC interval. This multiple is defined in the configuration ADF as `cycles_per_tie_point` (see section 6.8.3.1 of RD-11). If the time increment is chosen to be 8 SCANSYNC intervals (2.4 seconds) then the along-track sampling interval corresponds to 16 scan traces,



or approximately 16 km. In the following this sampling interval will be denoted by Δt . The orbit propagator is used to calculate the latitude and longitude at each tie point, together with the azimuth of the ground track at the point. Adding 90° to this gives the azimuth of the normal section orthogonal to the ground track, which is the line of constant Y, through the point.

Then for each point in turn, a direct geodetic formula (Section 4.4.4.9.3) is used to compute the latitudes and longitudes of a series of points equally spaced X, along this normal section.

4.4.3.1.1 Along-track look-up tables

The look-up table defining points on the ground track at equal time interval spacing is generated as follows. We define a series of times $t[i]$, $i = 0, 1, 2, \dots$

$$t[0] = T_0,$$

$$t[i] = t[i-1] + \Delta t, \quad i = 1, 2, 3, \dots$$

Here the time origin T_0 represents the start time of the output product, and will be the time of the ascending node crossing for consolidated products.

The orbit propagator is then used to calculate the position and velocity of the sub-satellite point at each time $t[i]$. If the latitude and longitude of the sub-satellite point at time $t[i]$ are $lat[i]$ and $long[i]$ respectively, then given $lat[i-1]$, $long[i-1]$, $lat[i]$ and $long[i]$, the length of the line segment $ds[i]$ between points $i-1$ and i can be derived (Section 4.4.4.9.3). Given $ds[i]$, the along-track distance $s[i]$ corresponding to the point at $t[i]$ can be calculated by

$$s[0] = 0.0$$

$$s[i] = s[i-1] + ds[i], \quad i = 1, 2, \dots$$

This is by definition the Y co-ordinate of the tie point.

4.4.3.1.2 Reference Grid of Image Co-ordinates

Starting with these along-track latitudes and longitudes, the latitude and longitude of every set of across-track points corresponding to the along track grid points are calculated. In practice, so as to ensure that the derived co-ordinates faithfully represent true distance on the ellipsoid, we will use a direct geodetic formula (Section 4.4.4.9.3) for this calculation. However, to illustrate the principle of the calculation we will first show how it would be done for a spherical Earth. The full formulae will then be given in a later section.

Consider the point $[i]$ on the ground track. If the velocity components of the sub-satellite point at this point in the Eastern (latitude or lambda) and Northern (longitude or phi) directions are $velam[l]$ and $vephi[l]$ respectively, then the azimuth (measured clockwise from North) of the ground track is

$$\beta' = \text{atan2}\{velam[l], vephi[l]\}.$$

(Note that because the orbit is retrograde β' will be a negative angle). The azimuth of the line orthogonal to the ground track is then $A = \beta' + \pi/2$. Now consider the spherical triangle NQG in Figure 4-2

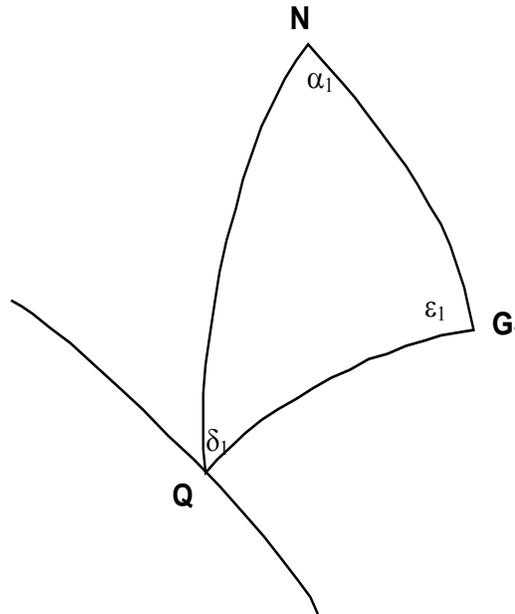


Figure 4-2.

In Figure 4-2, Q is the current ground track point indexed by i , N is the north pole, and G_1 is the next grid point, L km from Q along the great circle at azimuth $\delta_1 = \beta' + \pi/2$. In the triangle formed by these three points, the following quantities are known:

arc QN the co-latitude of Q)

The azimuth angle δ_1

arc QG (L/R radians by definition, where R is the radius of the Earth.)

Thus two sides and an included angle are known, and the remaining quantities α_1 , ϵ_1 , and arc NG_1 can be calculated by standard techniques. Then the latitude of G_1 is given by $90^\circ - \text{arc } NG_1$, and its longitude by (longitude of Q + α_1). This process can be iterated to give further points G_j along the extended arc QG_1 . The angle $\delta_{j+1} = \pi - \epsilon_j$ is the azimuth of the great circle of constant Y at G_j . The procedure can then be repeated on the other side of the track. The along-track coordinate of all the points in the across-track direction is the same as that of the sub-satellite point through which the orthogonal great circle is drawn (by definition).

In practice the triangle NQG is drawn on the ellipsoid, and more complex methods are needed to solve it. However, this does not alter the basic principle of the method. The full equations are given in Section (4.4.4.9).

4.4.3.1.3 Comments

The process in which an off-track point $G[i, j+1]$ is derived from the adjacent point $G[i, j]$ could be replaced by one in which each point was derived directly from the on-track point. This would eliminate the possibility of cumulative errors. It would also generalise easily to the case of ellipsoidal geometry.

The process of generating the off-track grid points essentially reduces to the problem of determining the coordinates of the end-point of a line segment, of length L and azimuth A , given the starting co-ordinates (latitude and longitude). This is a standard problem in geodesy and surveying, and established techniques from these fields can be used to extend the computation to the ellipsoid.



4.4.4 Geo-referencing instrument samples

The process of geolocation aims to define the co-ordinates on the Earth's surface corresponding to the centre of each scan pixel. The required co-ordinates are the latitude and longitude of the pixel on the reference ellipsoid. Ortho-geolocation can then be derived from knowledge of the Digital Elevation Model.

The co-ordinates of the scan pixel correspond to the intersection with the reference ellipsoid of the line of sight for a given detector as it leaves the scan mirror. The problem, then, is to determine this point of intersection, given the satellite position and attitude and the orientation of the scan mirror.

The derivation proceeds in two main stages. First, from knowledge of the angle through which the scan mirror has rotated, we determine the direction cosines of the line of sight relative to a frame of reference fixed in the satellite; then we use the attitude steering law of the satellite and knowledge of its position in its orbit to relate the direction to the Earth-fixed frame of reference. In practice it is expected that the second stage will be accomplished using CFI software for orbit generation and target location similar to that used for AATSR processing. These calculations must be repeated for each pixel for which geolocation is required.

4.4.4.1 Frames of reference

First of all we define the reference frames used in the calculations. All the frames of reference are defined to be right-handed.

Reference [RD-4] defines a number of Sentinel-3 related frames of reference, in particular the Spacecraft Reference Frame and the SLSTR Reference Frame. In addition, we will require additional frames related to the SLSTR scan geometry, and these are also described below. The transformations between these reference frames will then be expressed as a series of elementary rotations.

In effect, relative to the (A)ATSR configuration, the SLSTR instrument flies backwards, with the inclined view following the nadir view. The active portions of the two scans retain opposite curvatures, so the nadir view is concave backwards rather than forwards.

The standard frames of reference defined in the Sentinel 3 [AD-3] and SLSTR [RD-2] documentation differ from those used for ENVISAT and AATSR. Frames of reference in the Sentinel documents are defined so that the Y axis points in the across-track direction, in contrast to the convention used for (A)ATSR, in which the X axis was directed across-track. Thus the equivalent AATSR and SLSTR reference frames are related by a rotation through 90 degrees about their common Z axes. Here we will adopt the Sentinel-3 definitions, at least as far as the orientation of each frame is concerned. Strictly we require the origin of each reference frame to coincide with the centre of the scan mirror, rather than the frame-specific origins defined in [RD-2]. In practice any error, in terms of displacement on the surface, is presumably negligible.

4.4.4.2 The yaw steering frame

This frame is defined in Appendix A of [AD-3]. It is a right-handed Cartesian reference frame having its origin at the centre of mass of the satellite. The Z axis points towards the local geodetic nadir, and the X axis is parallel to, and in the same direction as, the ground trace velocity of the sub-satellite point. The Y axis then completes the right-handed set. This frame represents the nominal attitude of the satellite when in yaw steering mode; following [AD-3] co-ordinates in this system will be denoted by the subscript *yst*.

4.4.4.3 The spacecraft reference frame

This is defined in [RD-4] in terms of the spacecraft structure and the launch vehicle interface plane. However for our present purposes it can be defined as a Cartesian frame of reference, rigidly fixed in the spacecraft and oriented so that in the nominal flight configuration, the Z axis will point to geodetic nadir, and the X axis will be parallel to the satellite roll axis and directed backwards relative to the orbital velocity



vector. The Y axis then completes the right-handed set. Co-ordinates in this system will be denoted by the subscript s .

4.4.4.4 The platform reference frame

This frame is fixed relative to the satellite, and is obtained from the spacecraft reference frame by a rotation about 180 degrees about the common Z axes so that the X axis points forward. It is defined for convenience in specifying line of site azimuth and elevation; it is nominally parallel to the yaw steering frame when the satellite is operating in yaw steering mode. Co-ordinates in this system will be denoted by the subscript p .

4.4.4.5 The instrument reference frame

The instrument frame of reference is then a Cartesian frame having the Z axis directed at nadir, in the normal flight configuration. The X axis is then orthogonal to this, in the plane containing the two scan axes, and directed from the centre point of the nadir scan towards the centre point of the oblique scan, that is to say, backwards relative to the direction of flight. The Y axis then completes the right-handed set. It will be denoted by the subscript b .

With this definition the Y axis points across-track, and corresponds to the X axis of the corresponding frame of reference used in discussion of the (A)ATSR instruments. In order to retain the algebraic link between the AATSR and SLSTR, we will measure scan angles from the Y axis.

For present purposes we can identify this frame with the SLSTR reference frame defined in [RD-2], apart from a small displacement in the origin which is not material to the geolocation precision. (We actually want the origin of the frame to correspond to the centre of the scan mirror).

4.4.4.6 The scan reference frames

We then define scan frames for the nadir and inclined views. In each case the scan frame is defined as the frame whose Z axis lies along the axis of the scan, and whose Y axis is identical to that of the instrument frame defined above. The X axis then completes the right-handed set.

With this definition, the nadir view scan frame is obtained from the instrument frame by a rotation about the common Y axes through the angle κ' necessary to bring the Z axis into line with the axis of the nadir scan. This frame will be denoted by the subscript a .

However, it is immediately clear that since the oblique scan rotates about a vertical axis, the oblique scan frame as we have defined it is identical to the instrument frame.

The viewing directions are initially defined with respect to these frames as follows. In each case we assume that the viewing direction rotates in a positive sense about the Z_a axis, to which it is inclined at angle κ . Thus the scan on the surface is traced in a clockwise direction, as seen from above.

We can then express the line of sight as a vector in the scan frame, and transform it into the instrument frame, and then into the satellite frame, taking into account any misalignment between the frames. The line of sight depends on the scan angle and, for the SLSTR design, on the position of the detector element relative to the focal point of the channel, as we will discuss in a later section.

4.4.4.7 Satellite transformations

We then define linear transformations $M_{ab}(-\kappa')$, $M_z(\zeta)$, $M_y(\zeta)$, $M_x(\xi)$, M_{ps} as follows:

$M_{ab}(-\kappa')$ converts the components of a vector defined in X_a , Y_a , Z_a to those of the same vector defined with respect to the baseplate axes X_b , Y_b , Z_b .



$M_z(\zeta)$, $M_y(\eta)$, $M_x(\xi)$, applied in turn, convert the components of a vector defined in X_b , Y_b , Z_b to those of the same vector defined with respect to the spacecraft axes X_S , Y_S , Z_S .

M_{y_s} converts the components of a vector defined in X_S , Y_S , Z_S to those of the same vector defined with respect to the yaw steering frame X_{yst} , Y_{yst} , Z_{yst} .

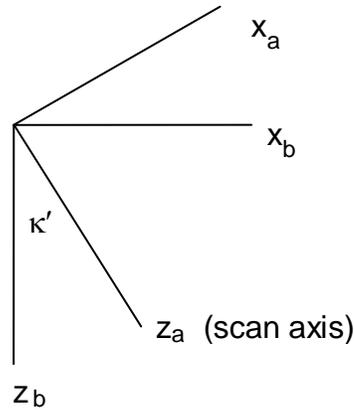


Figure 4-3

The equations of the transformations are as follows.

Suppose that x_a , y_a , z_a are the components of a vector \mathbf{x} relative to the nadir scan reference frame, and that x_b , y_b , z_b are the components of the same vector relative to the baseplate reference frame. $M_{ab}(-\kappa')$ is a rotation of $-\kappa'$ about the common Y_a , Y_b axes (Figure 4-3), and so the components are related by [actually k-prime]

$$\begin{aligned} x_b &= x_a \cos \kappa' + z_a \sin \kappa' \\ y_b &= y_a \\ z_b &= -x_a \sin \kappa' + z_a \cos \kappa' \end{aligned} \tag{eq 4.4-1}$$

Therefore

$$M_{ab}(-\kappa') = \begin{pmatrix} \cos \kappa' & 0 & \sin \kappa' \\ 0 & 1 & 0 \\ -\sin \kappa' & 0 & \cos \kappa' \end{pmatrix} \tag{eq 4.4-2}$$

Matrices $M_z(\zeta)$, $M_y(\eta)$, $M_x(\xi)$, represent elementary rotations of ζ , η , ξ about the z , y and x axes respectively. The transformation between the instrument and satellite frames is built up out of these elementary transformations. We adopt the convention that the rotations are to be applied in that order to give the total transformation to the platform frame. Strictly speaking the matrices representing elementary rotations about different axes do not commute, and so we should specify the order in which they are to be applied. In practice the angles are sufficiently small that any errors from this source are small in relation to the overall attitude error budget and the matrices can be regarded as commuting to a sufficient accuracy.

M_z represents a rotation about the axis Z_b to give a new intermediate frame of reference, in which the components of the vector \mathbf{x} are x'_b , y'_b , z'_b . The components of \mathbf{x} in the new basis are (Figure 4-4)



$$\begin{aligned}x'_b &= x_b \cos \zeta + y_b \sin \zeta \\y'_b &= -x_b \sin \zeta + y_b \cos \zeta \\z'_b &= z_b\end{aligned}\tag{eq 4.4-3}$$

and so

$$M_z(\zeta) = \begin{pmatrix} \cos \zeta & \sin \zeta & 0 \\ -\sin \zeta & \cos \zeta & 0 \\ 0 & 0 & 1 \end{pmatrix}\tag{eq 4.4-4}$$

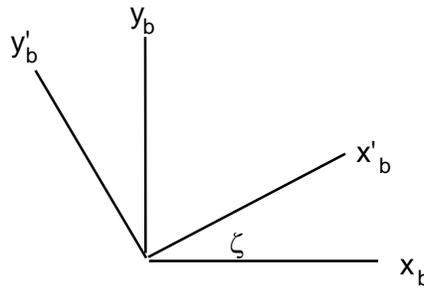


Figure 4-4

The matrix M_y represents a rotation about the transformed axis y'_b to give a new intermediate frame of reference (Figure 4-5) in which the components of \mathbf{x} are x''_b, y''_b, z''_b given by

$$\begin{aligned}x''_b &= x'_b \cos \eta - z'_b \sin \eta \\y''_b &= y'_b \\z''_b &= x'_b \sin \eta + z'_b \cos \eta\end{aligned}\tag{eq 4.4-5}$$

Therefore the matrix M_y is

$$M_y(\eta) = \begin{pmatrix} \cos \eta & 0 & -\sin \eta \\ 0 & 1 & 0 \\ \sin \eta & 0 & \cos \eta \end{pmatrix}\tag{eq 4.4-6}$$

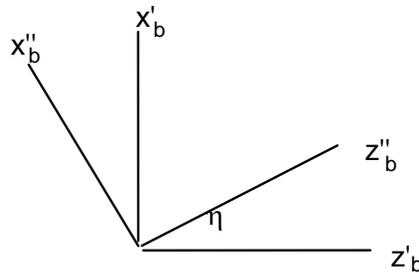


Figure 4-5

M_x represents the rotation about the transformed x''_b axis to the satellite frame (Figure 4-6). The components of \mathbf{x} in this basis are x_p, y_p, z_p given by

$$\begin{aligned} x_p &= x''_b \\ y_p &= y''_b \cos \xi + z''_b \sin \xi \\ z_p &= -y''_b \sin \xi + z''_b \cos \xi \end{aligned} \tag{eq 4.4-7}$$

and so

$$M_x(\xi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \xi & \sin \xi \\ 0 & -\sin \xi & \cos \xi \end{pmatrix} \tag{eq 4.4-8}$$

The above equations can be regarded as defining the misalignment angles.

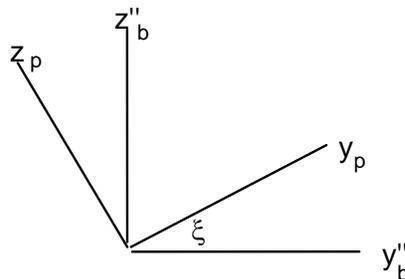


Figure 4-6

4.4.4.8 Treatment of multiple detectors

In general, the first stage of geolocation involves the application in turn of the above transformations to the line of sight vector expressed in the scan reference frame. This gives the vector expressed in the spacecraft reference frame; this is then used as a parameter of the CFI target function to define the instrument pixel position. We therefore need an expression for the initial line of sight.

In the case of the (A)ATSR instruments, the field stop at the prime focus of the telescope was imaged onto a single detector. Consequently the position of the pixel on the ground was given by the intersection with the surface of the line of sight from the focal point. It is very simple to write down the direction of the line of sight in this case.



Let the rotation angle of the scan mirror be φ , defined to be zero when the scan mirror normal lies in the Y - Z plane of its respective scan reference frame. With respect to the scan reference frame, the direction cosines of the line of sight are then

$$\lambda = -\sin \kappa \sin \varphi$$

$$\mu = \sin \kappa \cos \varphi$$

$$\nu = \cos \kappa$$

eq 4.4-9

where κ is the apex semi-angle of the scan cone. (The negative sign appears because we have defined the scan angle in a way that is consistent with the definition used in the AATSR documentation.)

In the case of SLSTR, each channel will have more than one detector in the FPA, and these detectors will be displaced from the position of the focal point. More than one pixel will be imaged simultaneously (two in the case of the thermal channels, eight in the case of the short-wave and visible channels). For each detector element, the position of the corresponding pixel on the ground will be the intersection with the surface of the line of sight from the detector. In order to treat this case, we must replace the equations (4.4.9), which define the line of sight from the focus, by expressions which take account of the displacement of the detector from the focus. These will be different for each distinct detector element of a channel.

Let us suppose that the detector element under consideration is displaced from the focal point by a vector element (dx, dy) relative to the scan axes. (Strictly this refers to the point in the focal plane of the telescope that is imaged onto the detector element; we do not need to take account of the detailed geometry of the FPA optics).

If F represent s the focal length of the telescope, the direction in the scan frame, after reflection, from which rays will be focussed onto the detector element at (dx, dy) is given by the vector $\mathbf{f} = (dx, dy, F)$.

The corresponding unit vector is therefore given by $\boldsymbol{\tau} = \text{col}\{dx \ dy \ F\}/|\mathbf{f}|$, where $|\mathbf{f}| = \sqrt{dx^2 + dy^2 + F^2}$ is the modulus of the vector \mathbf{f} . The relationship between this vector and the reflected ray direction follows from the equation of reflection

$$\boldsymbol{\tau}' + \boldsymbol{\tau} = 2(\boldsymbol{\tau} \cdot \mathbf{n})\mathbf{n} .$$

eq 4.4-10

This can be rewritten as

$$\boldsymbol{\tau}' = 2(\boldsymbol{\tau} \cdot \mathbf{n})\mathbf{n} - \boldsymbol{\tau} = 2(\tau_x n_x + \tau_y n_y + \tau_z n_z)\mathbf{n} - \boldsymbol{\tau} .$$

eq 4.4-11

This is a linear function of $\boldsymbol{\tau}$ and so can be expressed by a matrix transformation. In terms of the components of \mathbf{n} , the expression becomes

$$\boldsymbol{\tau}' = \begin{pmatrix} 2n_x^2 - 1 & 2n_y n_x & 2n_z n_x \\ 2n_x n_y & 2n_y^2 - 1 & 2n_z n_y \\ 2n_x n_z & 2n_y n_z & 2n_z^2 - 1 \end{pmatrix} \boldsymbol{\tau}$$

eq 4.4-12

The instantaneous direction of the mirror normal \mathbf{n} is given by

$$n_x = -\sin \varphi \sin i$$

$$n_y = \cos \varphi \sin i$$

$$n_z = \cos i$$

eq 4.4-13

In this equation φ is the scan angle as before, and i is the angle of incidence of the optical axis on the mirror. (Note this has the same form as (1), but with i in place of the cone angle κ .)



Substituting these expressions in the matrix equation we find

$$\boldsymbol{\tau}' = \begin{pmatrix} 2\sin^2\varphi\sin^2i-1 & -2\cos\varphi\sin\varphi\sin^2i & -2\sin\varphi\sin i\cos i \\ -2\cos\varphi\sin\varphi\sin^2i & 2\cos^2\varphi\sin^2i-1 & 2\cos\varphi\sin i\cos i \\ -2\sin\varphi\sin i\cos i & 2\cos\varphi\sin i\cos i & 2\cos^2i-1 \end{pmatrix} \boldsymbol{\tau} \quad \text{eq 4.4-14}$$

This rather complicated-looking matrix can in fact be factorised into the following expression:

$$\boldsymbol{\tau}' = M_{ac}^{-1}M_{cm}^{-1} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} M_{cm}M_{ac}\boldsymbol{\tau} \quad \text{eq 4.4-15}$$

In this expression the quantities M_{cm} and M_{ac} represent elementary rotations through φ and i respectively as follows:

$$M_{ac} = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}; M_{cm} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{pmatrix} \quad \text{eq 4.4-16}$$

It is easy if somewhat laborious to verify this factorisation.

Note that if the initial vector is a unit vector directed outward along the Z_a axis, i.e. if $\boldsymbol{\tau} = \text{col}\{0, 0, 1\}$, then the outward line of sight from the mirror is

$$\boldsymbol{\tau}' = \begin{pmatrix} -\sin\varphi\sin 2i \\ \cos\varphi\sin 2i \\ \cos 2i \end{pmatrix} \quad \text{eq 4.4-17}$$

This represents the ATSR geometry, and is identical to (2.4.9) with $\kappa = 2i$.

In the case of the oblique scan, the above equations give the direction cosines of the line of sight relative to the common scan and instrument frames. In the case of the nadir scan, we must transform the line of sight vector into the instrument frame.

Given the direction of the line of sight as calculated above, the CFI software can be used to determine the location (latitude and longitude) of the pixel on the Earth.

4.4.4.9 Geometry on the Ellipsoid

We have seen above that the Level 1b output product grid is uniformly sampled in the co-ordinates X and Y. In order to relocate the pixels on this grid it is therefore necessary to transform the latitude and longitude of each pixel into the X and Y co-ordinate system. In order that the X - Y co-ordinate system gives a true representation of distances on the surface of the Earth, the transformation equations relating latitude and longitude to X and Y must take account of the geometry on the ellipsoid. Therefore before we present the transformation equations, we set out some relevant geodetic theory.

4.4.4.9.1 Some Relationships on the Ellipsoid

This section gives for reference a number of standard geometrical relationships relating to the ellipsoid. Some are given without proof, but can be found in texts such as References [RD-13].

The reference ellipsoid is the ellipsoid of revolution formed by rotating an ellipse about its semi-minor axis (which will coincide with the axis of rotation of the Earth). The cross-section of the surface in any plane



containing the axis of rotation will be an ellipse. If we define axes x and y to coincide with, respectively, the semi-major and semi-minor axes of this ellipse, we can write its equation in the standard form

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (4.4.18)$$

where a is the semi-major axis and b is the semi-minor axis. The ratio $(a-b)/a$ is known as the flattening, f . It is related to the eccentricity e by

$$(1-f)^2 = (1-e^2) \quad (4.4.19)$$

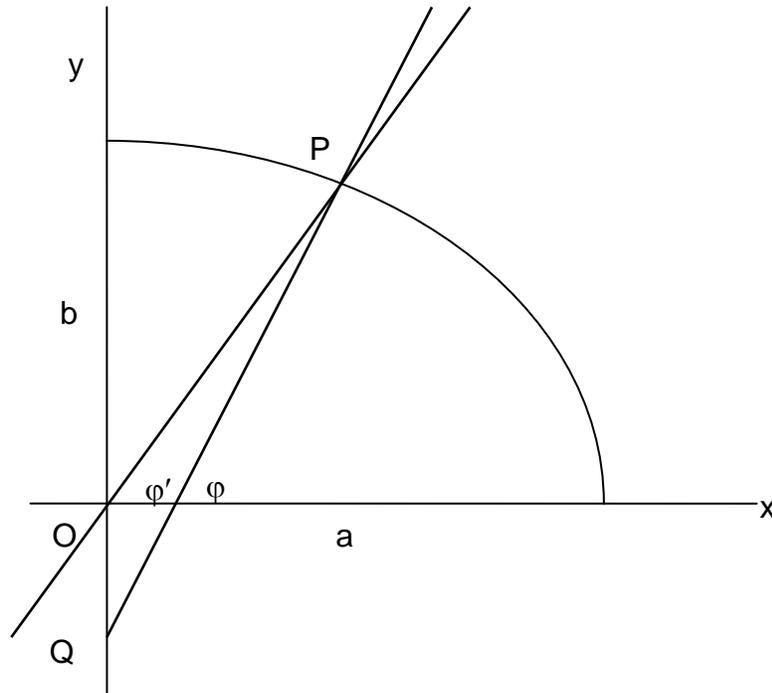


Figure 4-7

Figure 4-7 shows a point $P(x', y')$ on the surface. The angle ϕ' is the geocentric latitude of P given by

$$\tan \phi' = y'/x' \quad (4.4.20)$$

The line PQ is the normal to the surface at P . The angle ϕ that this makes with the x axis (or the equatorial plane) is the geodetic latitude of P ;

$$\tan \phi = 1 / \left(\frac{dy}{dx} \right)_P = \frac{y'}{(1-f)^2 x'} \quad (4.4.21)$$

Thus

$$\tan \phi' = (1-f)^2 \tan \phi = (1-e^2) \tan \phi \quad (4.4.22)$$

The radius of curvature of the surface in the meridian plane at point P is given by

$$R = \frac{a(1-e^2)}{(1-e^2 \sin^2 \phi)^{3/2}} \quad (4.4.23)$$



This equation can be derived fairly directly from the equation for the ellipse (Equation 4.4.18). The radius of curvature in the vertical plane normal to the meridian plane (in Figure 4-7, the plane containing PQ and normal to the plane of the diagram) is

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \quad (4.4.24)$$

This is sometimes termed the radius of curvature in prime vertical. The point Q is the common intersection of the surface normals at points on the parallel of latitude through P, and the length PQ can be shown to equal the normal radius of curvature N . Thus the centre of curvature in prime vertical lies on the minor axis of the ellipse. Note that the alternative notation ν , ρ is often used in the literature for the radii of curvature N , R respectively, and we shall use this notation below.

The equation of the line PQ is

$$(y - y') = (x - x') \tan \varphi \quad (4.4.25)$$

and its intercept Q at $x = 0$ is

$$y = y' - x' \tan \varphi = -y' e^2 / (1 - e^2) \quad (4.4.26)$$

Thus the projection of PQ onto the y axis is

$$N \sin \varphi = y' + y' e^2 / (1 - e^2) = y' / (1 - e^2) \quad (4.4.27)$$

and finally we have an expression for the coordinates of P in terms of N and φ

$$x' = N \cos \varphi, \quad y' = N(1 - e^2) \sin \varphi \quad (4.4.28)$$

Define a right-handed set of Cartesian axes X , Y , Z with origin at the centre of the Earth, with Z directed northwards along the axis of rotation, and with X in the plane of the prime meridian. The XY plane is therefore the plane of the equator. If Figure 4-7 represents the meridian plane at longitude λ , the X , Y , Z coordinates of a point (x, y) in the plane are

$$\begin{aligned} X &= x \cos \lambda \\ Y &= x \sin \lambda \\ Z &= y \end{aligned} \quad (4.4.29)$$

The geocentric Cartesian coordinates (X, Y, Z) of a point T at a height h vertically above P are therefore

$$\begin{aligned} X &= (N + h) \cos \varphi \cos \lambda \\ Y &= (N + h) \cos \varphi \sin \lambda \\ Z &= [N(1 - e^2) + h] \sin \varphi \end{aligned} \quad (4.4.30)$$

Length and azimuth on the ellipsoid

We also require formulae for the length and azimuth of the curves joining any two points P_1 and P_2 on the ellipsoid. Firstly it is necessary to be clear as to what is meant, since any of three different curves may be meant. The geodesic is the unique curve of shortest length between the two points P_1 and P_2 ; however, many of the standard formulae refer not to the geodesic but to one of the two normal sections between the points.



The geodesic does not lie in a plane, nor does its azimuth α coincide with the bearing of point P_2 that would be measured with a levelled theodolite at point P_1 , although the deviations will be very small for short lines. There are an infinite number of plane sections through the ellipsoid including points P_1 and P_2 . In particular we can consider two; the intersection of the ellipsoid with the plane containing point P_2 and the normal to the surface at point P_1 , which we may call S_1 , and the intersection with the plane containing point P_1 and the normal to the surface at point P_2 , which we may call S_2 . These are the normal sections, and their azimuths, measured at the points at which the plane section contains the normal to the surface, are the normal section azimuths. Thus the angle between the curve S_1 and the meridian plane through point P_1 is the normal section azimuth of point P_2 measured from point P_1 , and this angle is the bearing of point P_2 that would be measured with a levelled theodolite at point P_1 .

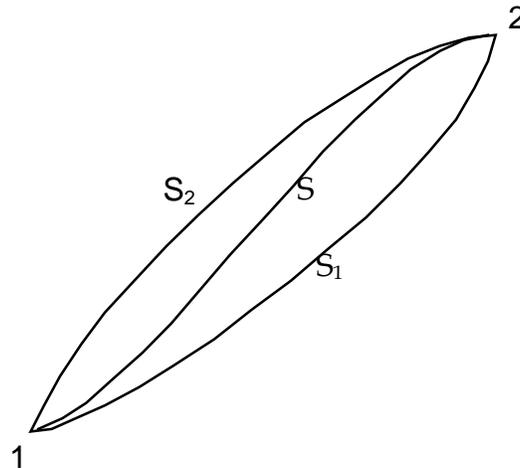


Figure 4-8

The geodesic S lies between these two arcs S_1 and S_2 , somewhat as shown in Figure 4-8. The angles between the curves are very small; if we consider azimuths measured at point P_1 we have

$$\alpha_2 - \alpha_1 = 3(\alpha_2 - \alpha) \cong 00042 \text{ arc sec} \quad (4.4.31)$$

for line lengths $L = 10 \text{ km}$. The difference increases quadratically with L .

It is possible to give an exact expression for the normal section azimuth if the latitude and longitude of the two points are known. To simplify the notation, rename the two points P and Q , and let their geodetic latitude and longitude be (φ_1, λ_1) and (φ_2, λ_2) respectively. Consider the sphere of radius $O'P$ centred at O' , which is the intersection of the normal through P with the axis of rotation. This sphere is tangent to the ellipsoid at the parallel of latitude through P , but does not otherwise coincide with it. Consider the spherical triangle NPQ' drawn on this sphere (Figure 4-10), where N is the intersection of the axis with the sphere, and Q' is the intersection of the line $O'Q$ with this sphere. As we have defined it, the plane $O'PQ'$, which also contains point Q , is the normal section at P containing Q , and so the desired normal section azimuth of Q measured from P is the angle at P (NPQ'). In the triangle, the angle at N is known, since it is given by the difference in the longitudes of Q and P , as are the sides NP and NQ' , so that the triangle can be solved for angle P . NP is the geodetic co-latitude of point P . The calculation of the arc NQ' is more complicated.

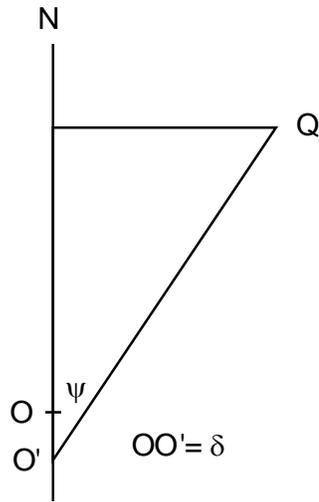


Figure 4-9

Suppose O is the centre of the Earth. We know that the z coordinate of O' is

$$\delta = -e^2 N_1 \sin \varphi_1.$$

At the same time the z coordinate of Q is (Equation 4.4.30)

$$N_2 (1 - e^2) \sin \varphi_2.$$

Thus in Figure 4-9, which shows the meridian plane through Q, the length O'Z (where Z is the projection of Q on the polar axis) is

$$N_2 (1 - e^2) \sin \varphi_2 + e^2 N_1 \sin \varphi_1 \tag{4.4.32}$$

(The subscripts 1 and 2 refer to the points P and Q respectively.) The length ZQ is

$$N_2 \cos \varphi_2$$

and so we have

$$\tan \psi = \cos \varphi_2 / ((1 - e^2) \sin \varphi_2 + e^2 (N_1/N_2) \sin \varphi_1), \tag{4.4.33}$$

from which the angle ψ can be determined.

In Section 4.4.4.9.4 we outline a general approach to the solution of a spherical triangle in which two sides and the included angle are known. By application of this method to the triangle PNQ' (Figure 4-10) we can obtain explicit expressions for $\sin A$ and $\cos A$ in terms of known quantities and of the unknown χ . Angle A is the azimuth of the normal section PQ at P.

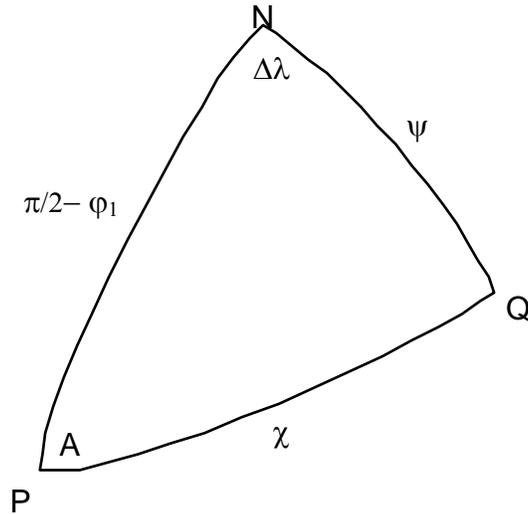


Figure 4-10

From the sine rule we get

$$\sin A = \sin \Delta\lambda \sin \psi \operatorname{cosec} \chi. \quad (4.4.34)$$

Using two applications of the cosine formula, we can express both χ and ψ in terms of known quantities:

$$\cos \chi = \cos(\pi/2 - \varphi_1) \cos \psi + \sin(\pi/2 - \varphi_1) \sin \psi \cos \Delta\lambda$$

$$\cos \psi = \cos(\pi/2 - \varphi_1) \cos \chi + \sin(\pi/2 - \varphi_1) \sin \chi \cos A$$

We can now eliminate $\cos \chi$ from these equations. Multiply the first equation by $\sin \varphi_1$;

$$\cos \chi \sin \varphi_1 = \sin^2 \varphi_1 \cos \psi + \cos \varphi_1 \sin \varphi_1 \sin \psi \cos \Delta\lambda$$

and re-order the second;

$$\cos \chi \sin \varphi_1 = \cos \psi - \cos \varphi_1 \sin \chi \cos A$$

The left-hand sides are equal, and so we can equate the right-hand sides. Therefore

$$\cos \varphi_1 \sin \chi \cos A = \cos \psi (1 - \sin^2 \varphi_1) - \cos \varphi_1 \sin \varphi_1 \sin \psi \cos \Delta\lambda$$

Divide this equation by $\cos \varphi_1 \sin \chi$;

$$\cos A = [\cos \psi \cos \varphi_1 - \sin \varphi_1 \sin \psi \cos \Delta\lambda] \operatorname{cosec} \chi.$$

Therefore

$$\begin{aligned} \cot A &= \operatorname{cosec} \Delta\lambda [\cot \psi \cos \varphi_1 - \sin \varphi_1 \cos \Delta\lambda] \\ &= \sin \varphi_1 \operatorname{cosec} \Delta\lambda [\cot \psi / \tan \varphi_1 - \cos \Delta\lambda] \end{aligned} \quad (4.4.35)$$

where

$$\cot \psi = \Lambda_{12} \tan \varphi_1 \quad (4.4.36)$$

and

$$\Lambda_{12} = (1 - e^2) \frac{\tan \varphi_2}{\tan \varphi_1} + e^2 \left(\frac{v_1}{v_2} \right) \frac{\tan \varphi_2}{\tan \varphi_1} \quad (4.4.37)$$

from Equation (4.4.33). This is Cunningham's Azimuth formula (Bomford [RD-13]).

An exact equation for the length of the normal section is given by Bomford [RD-13]. Various approximate forms are also available, and will be discussed below.



4.4.4.9.2 Direct and Inverse Formulae

In order to define the SLSTR image grid (to determine the latitude and longitude of a general grid point), we will first define the latitudes and longitudes of a series of points on the Sentinel 3 ground track. The co-ordinates of the other grid points will then be defined on the assumption that each lies on a line through a ground track point, at a given azimuth and at a given distance (the grid X co-ordinate) from the ground track point.

This is an instance of a general problem that arises in surveying and geodesy, to find the co-ordinates of a point at a known distance, along a line at a known azimuth, from a point whose co-ordinates are known. If the two points are P_1 and P_2 , and their geodetic latitude and longitude are respectively (φ_1, λ_1) , (φ_2, λ_2) , the problem is to find (φ_2, λ_2) given the distance L between P_1 and P_2 , and the azimuth measured from North at P_1 , together with the latitude and longitude of P_1 . A formula that gives (φ_2, λ_2) in terms of the known quantities is known as a 'direct formula'.

Conversely, to determine the X and Y co-ordinates of a point given its latitude and longitude, we require the inverse relationship, that gives the length and azimuth of the line between two points on the ellipsoid, as a function of the co-ordinates (latitude and longitude) of the two points.

Many geodetic formulae exist, of varying degrees of complexity and accuracy. Several are given by Bomford ([RD-13] and earlier editions of that work). Formulae differ according as whether they use geodesic or normal section distances and azimuths. Some formulae (e.g that of Vincenzi) are expressed in terms of the geodesic distance between the points, and the geodesic azimuths. For any modification of the AATSR regridding scheme, we desire to maintain consistency with other ENVISAT instruments, and in particular to ensure that the line of constant y should lie in the X - Z plane of the satellite, and this clearly defines a normal section at the sub-satellite point. We thus require normal section formulae. It is true that for sufficiently short lines the difference between the lengths of the geodesic and the normal section is negligible, but it might still be necessary to take account of the azimuth differences to maintain control of errors. Of the formulae cited by Bomford [RD-13], the two adopted for AATSR processing were those of Clarke and Robbins; Robbins's formula is essentially the inverse of that of Clarke, and we consider it first.

Suppose that the co-ordinates (latitude and longitude) of two points P and Q are known. Given these quantities, the normal section azimuth of Q measured at P can be computed using Cunningham's azimuth formula (above). In the same spherical triangle (Figure 4-10) the angular distance χ (arc PQ measured at the centre of curvature in prime vertical at P , or O' in Figure 4-9) can also be derived from Equation (4.4.34 - 36). The corresponding length on the auxiliary sphere centred at O' is $(v_1\chi)$. For small separations PQ it is clear that this length is very close to the desired length of the normal section PQ , so that the latter can be derived by the application of a small correction to χ . In effect this is what Robbins's formula does.

The radii of curvature in prime vertical at the points P , Q are as follows:

$$v_1 = a / (1 - e^2 \sin^2(\varphi_1))^{3/2} \quad (4.4.38)$$

$$v_2 = a / (1 - e^2 \sin^2(\varphi_2))^{3/2} \quad (4.4.39)$$

where a is the semi-major axis (the equatorial radius) of the Earth; cf. Equation (4.4.24). We can then calculate the angle ψ as in equation (4.4.33):

$$S_\psi = (1 - e^2) \sin \varphi_2 + (v_1/v_2) e^2 \sin \varphi_1 \quad (4.4.40)$$

$$C_\psi = \cos \varphi_2 \quad (4.4.41)$$

$$\psi = \arccot(S_\psi / C_\psi) \quad (4.4.42)$$



Calculate the geodetic correction coefficients

$$g = \varepsilon \sin \varphi_1 \cos \varphi_1 \cos A \quad (4.4.43)$$

$$h = \varepsilon \cos^2 \varphi_1 \cos^2 A \quad (4.4.44)$$

where

$$\varepsilon = e^2 / (1 - e^2) \quad (4.4.45)$$

The line segment between points P and Q is then given as a function of the angle χ :

$$L = v_1 \chi \left[1 - \frac{\chi^2}{6} h(1-h) + \frac{\chi^3}{8} g(1-2h) + \dots \right] \quad (4.4.46)$$

plus terms in higher powers of χ .

Clarke's direct formula, which we quote without proof, is then given as follows. Suppose we are given the latitude and longitude of point P (represented by subscript 1), and the length L of the arc PQ and the normal section azimuth A of point Q;

Calculate

$$r_2 = -\varepsilon \cos^2 \varphi_1 \cos^2 A \quad (4.4.47)$$

$$r_3 = 3\varepsilon(1 - r_2) \cos \varphi_1 \sin \varphi_1 \cos A \quad (4.4.48)$$

Calculate the radius in prime vertical at latitude φ_1 ,

$$v_1 = a / (1 - e^2 \sin^2(\varphi_1))^{1/2}. \quad (4.4.49)$$

Then

$$g = \frac{L}{v_1} \left[1 - \frac{r_2(1+r_2)}{6} \left(\frac{L}{v_1} \right)^2 - \frac{r_3(1+3r_2)}{24} \left(\frac{L}{v_1} \right)^3 \right] \quad (4.4.50)$$

Refer again to Figure 4-10. The quantity just calculated is the angular distance PQ' (χ in the diagram). Once this is known, the triangle can be solved by standard techniques for the longitude difference $\Delta\lambda$ and for the angle ψ . Finally the latitude of Q can be determined from the latter as follows.

$$\rho = 1 - \frac{1}{2} r_2 g^2 - \frac{1}{6} r_3 g^3 \quad (4.4.51)$$

$$S_2 = \cos \psi - e^2 \rho \sin \varphi_1 \quad (4.4.52)$$

$$C_2 = (1 - e^2) \sin \psi \quad (4.4.53)$$

$$\tan \varphi_2 = S_2 / C_2 \quad (4.4.54)$$

Equations (4.4.51) to (4.4.54) are essentially a variant of Cunningham's formula, but adapted to the case that v_2 (the radius of curvature at point Q) is not known.

How many terms is it necessary to take in the series (enclosed in square brackets) in Robbins's formula (4.4.46) to ensure a given degree of accuracy? Table 5.4.1 below gives an upper limit on the magnitude of each term, including some we have not quoted explicitly, for three different line lengths; 25 km



(corresponding to the interval between along-track points used in computing the tables of along track distance), 32 km (ditto for a larger granule size) and 275 km, corresponding to the maximum across-track distance computed during the generation of the reference grid.

Table 5.4.1. Magnitude of terms in Robbins's Formula. (Terms in mm.)

Term	L = 25 km	32 km	275 km
χ^2	0.427 mm	0.895 mm	568.0 mm
χ^3	1.25 e-3	3.37 e-3	18.37
χ^4	6.57 e-7	2.25 e-6	0.106
χ^5	3.21 e-9	1.41 e-8	5.69 e-3

Clearly it is sufficient for our purposes to take only the first two terms. Table 5.4.2 shows the similar terms in Clarke's formula Equation (4.4.50).

Table 5.4.2. Magnitude of terms in Clarke's Formula. (Terms in mm.)

Term	L = 25 km	32 km	275 km
$(L/v_1)^2$	0.427 mm	0.895 mm	568.0 mm
$(L/v_1)^3$	1.25 e-3	3.37e-3	18.37

4.4.4.9.3 Solution of spherical triangles

In order to make use of these equations, it is necessary to be able to solve spherical triangles and triangles drawn on the ellipsoid. In this and the following sections we outline the equations for doing this. The implementation of this is described in section 5.3.3.2.1.

Suppose ABC is a spherical triangle, and suppose that its sides, in angular measure, are designated by a, b, c according to the usual convention. Thus a represents the length of the side BC, opposite the angle A, and so on. If the sides b and c are known, together with the included angle A, the remaining quantities a, B and C, may be determined. The relations between the sides and angles of the spherical triangle are well known, and can be found in standard references.

Using these, we can proceed as follows:

(i) Firstly, an application of the cosine rule gives us

$$\cos a = \cos b \cos c + \sin b \sin c \cos A$$

(ii) The sine rule can be written

$$\sin A / \sin a = \sin B / \sin b = \sin C / \sin c$$

from which we can deduce, multiplying by $\sin a \sin b \sin c$ throughout

$$\sin b \sin c \sin A = \sin a \sin c \sin B = \sin a \sin b \sin C.$$

The LHS is known, so we can write

$$\sin a \sin c \sin B = \sin b \sin c \sin A$$

$$\sin a \sin b \sin C = \sin b \sin c \sin A$$

(iii) Further applications of the cosine formula to the sides b and c respectively give after rearrangement

$$\sin a \sin c \cos B = \cos b - \cos a \cos c$$

$$\sin a \sin b \cos C = \cos c - \cos a \cos b$$

All the quantities on the RHS of (iii) are given apart from $\cos b$, which is derived from (i).



Thus relations (ii) and (iii) give us expressions for the tangent or cotangent of the angles B and C, from which the angles themselves can be deduced.

$$B = \text{atan2}\{\sin a \sin c \sin B, \sin a \sin c \cos B\} = \text{atan2}\{\sin b \sin c \sin A, (\cos b - \cos a \cos c)\}$$

$$C = \text{atan2}\{\sin a \sin b \sin C, \sin a \sin b \cos C\} = \text{atan2}\{\sin b \sin c \sin A, (\cos c - \cos a \cos b)\}$$

The side a can be deduced directly from (i); but if its magnitude is small, there might be a loss of precision. In order to allow for this possibility, once B has been calculated it is possible to calculate sin a by a further application of the sine rule (ii);

$$\sin a = \sin b \sin A / \sin B$$

Thus using this equation the arctan function can be used.

$$a = \text{atan2}\{(\sin b \sin A / \sin B), (\cos b \cos c + \sin b \sin c \cos A)\}$$

4.4.4.9.4 Sign conventions

The algorithm to solve a spherical triangle just described is assumed that the sides of the triangle are less than 180° in magnitude. A triangle with this property, sometimes termed an Euler triangle, has the further property that its angles are also less than 180°. Thus the natural application of the algorithm is to the case that the sides a, b, c and angles A, B, C all have positive values between 0° and 180°.

In the application to the SLSTR processing, the algorithm is always applied to triangles whose sides are less than 180°, but it is not always the case that the included angle is positive. In particular, the routine is most frequently applied to triangles in which the apex at which the included angle (A) is defined coincides with the North pole of the Earth, and the angle A is defined by the difference of two longitudes. This longitude difference is always calculated so that its magnitude is less than 180°, but it may be negative. We should therefore discuss what happens if the algorithm is entered with the included angle A negative. This bears on the two questions of the sign convention of the azimuth, and of the consistency of the sign of azimuth angles defined at a point.

We assume throughout that the specified sides b and c are in the range 0° to 180°, so that sin b and sin c are positive. First suppose that A is specified as positive angle, in the range 0° < A < 180°. Then sin A is also positive, and the right-hand sides of the equations (ii) are positive. The angles B and C are unambiguously determined as to magnitude and quadrant by the equations (ii) and (iii) respectively on the assumption that the proportionality constant (sin a sin c) is positive; in the case under consideration this implies that the sines of B and C are positive. Thus B and C will lie in 0 to 180°. If a is then derived from the sine rule, a consistent positive value will be derived.

If now a negative value is substituted for A (i.e. if -180° < A < 0°), sin A is also negative and the right-hand sides, of the equations (ii) will be negative. Then both B and C will be found to have a negative sign (i.e. each will lie in -180° to 0°). If the sine rule is then used to determine sin a, it will be found to be positive, which is consistent with the implicit assumption that (sin a sin c) is positive. The resulting solution still satisfies the cosine law equations. Thus if the values a, b, c, A, B, C, satisfy the equations, so does the set a, b, c, -A, -B, -C in which the sign of each angle is reversed. This is the solution that will be obtained if the subroutine is entered with a negative argument for a.

The sine rule is consistent if the sign of an angle and its opposite side are simultaneously reversed. The cosine rule equations are all unchanged if the signs of the angles A, B and C are reversed.

It might be noted that there is an alternative self-consistent solution of the equations. The angle -A is equivalent to 360° - A, and the triangle ABC might be interpreted as the non-Eulerian triangle of sides 2π -



a, b, c. If we replace $\sin A$ by $\sin(360^\circ - A)$ and $\sin a$ by $\sin(2\pi - a)$, the signs of both are reversed. The sine rule is then consistent if $\sin B$ and $\sin C$ are both unchanged, while to maintain consistency of the cosine rule equations it is necessary to reverse the signs of $\cos B$ and $\cos C$ since they appear multiplied by $\sin A$. The corollary is that B and C must be replaced by $180 - B$ and $180 - C$ respectively, and these are the correct angles of the non-Eulerian triangle, measured inside the triangle. We do not get this solution if we enter the subroutine because it implicitly assumes that $\sin a$ is positive.

4.4.4.10 Calculation of Grid coordinates of Scan Pixel

The X and Y co-ordinates may in principle be calculated for any scan pixel, but in practice they are only directly calculated for a series of tie points, represented by every tenth pixel along each scan. The calculation makes use of the table of ground track point latitudes and longitudes described in Section 4.4.3.1.1.

Consider a pixel P. If a section is drawn through P to meet the ground track at a right angle in point X, the point X will lie between the tabular points Q[i] and Q[i+1], as shown schematically in Figure 4-11. The X co-ordinate of the pixel is then taken to be the distance PX of the pixel from the ground-track, measured along the section. The Y co-ordinate is determined from the distance, measured along the ground-track, between the i'th tabular point and the point X, which represents the difference between the y co-ordinate of X (and hence of P) and that of the i'th tabular point. Thus the distance Q[i]X is added to the Y co-ordinate of Q[i] (25i km) to give the y coordinate of P.

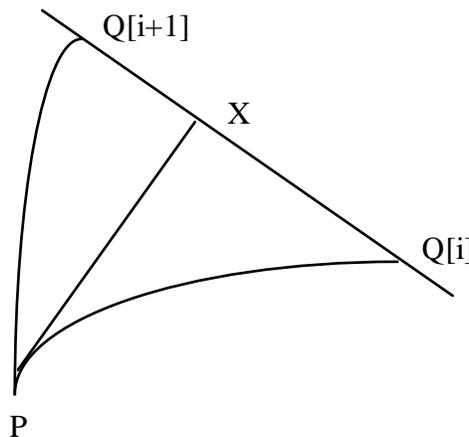


Figure 4-11

For the purposes of this calculation, the ground track between points Q[i] and Q[i+1] is regarded as approximated by the normal section between them. This is justified by the short length (16 km) of the segment; the ground track is not a geodesic, and cannot be approximated geodesic over long distances. The definition of the table of ground track points at 16 km intervals is essentially a means to by which the ground track is approximated by a piecewise continuous curve made up of 16 km segments.

4.4.4.10.1 Identification of the nearest tabular points

The first step is to identify the value of i corresponding to the interval within which the normal from P intersects the ground track. To be precise, this is the interval that contains a point X such that the normal section through X at right angles to the ground track contains the point P. The correct interval is determined by a test on the azimuths at the ground track points Q_i; the angle between the ground track and the PX varies continuously as the point X moves along the ground track, and therefore the correct interval is identified using the criterion that the angle of intersection passes through 90° within the interval.



Thus the intersection angle is calculated at the end-points of the interval. If the magnitude of angle $PQ[i]Q[i+1]$ is less than 90° , and that of $PQ[i+1]Q[i+2]$ is greater than 90° , then index i defines the correct interval.

To calculate the intersection angle at tabular point i , the azimuth of the ground track segment $Q[i]Q[i+1]$ is first determined, then the azimuth of the great circle from $Q[i]$ to point P . The intersection angle is given by the difference between the azimuths.

To determine the azimuth of the arc QP , the spherical triangle NPQ is solved, where N is the north pole of the Earth. If the general 'solve spherical triangle' routine (Section 5.3) is used, the length of the arc PQ and the azimuth of the arc at P are also determined as a by-product. The azimuth of the great circle segment is similarly determined by solution of the triangle $NQ[i]Q[i+1]$.

Consider Figure 4-12. The latitudes and longitudes of the three points P , Q_1 , Q_2 are known, and therefore an inverse formula can be used to derive the azimuths of P and Q_2 as seen from Q_1 ; in the diagram these are the angles ε and β respectively.

[One exceptional case may be accounted for; if the 180° meridian intersects the arc Q_1Q_2 , the equation above for the longitude difference will give a negative value close to -360° . In this case 360° (2π) may be added to the angle (although its sine and cosine will not change, and the 'solve spherical triangle' routine will give the same answer; the issue is one of precision rather than of the form of the equations.)]

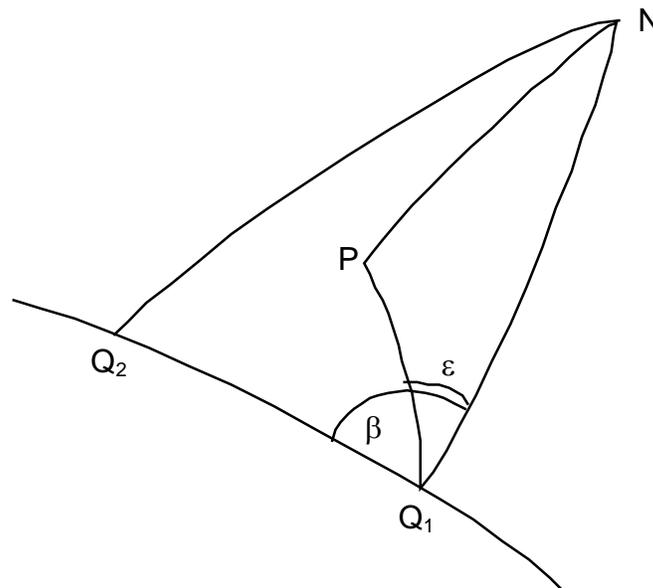


Figure 4-12

In accordance with the discussion of section 5.3.1.27, the angle β will come out in the range $0 \leq \beta < \pi$, and will represent the azimuth of the track at Q_1 , measured anticlockwise from north. Similarly ε will represent the azimuth of P seen from at Q_1 , according to the same sign convention. The difference $\beta - \varepsilon$ (reduced into the range $-\pi$ to π if necessary) represents the required intersection angle PQ_1Q_2 .

Thus starting from a specific tabular point i , the intersection angle at point $Q[i]$ is calculated. If its magnitude is less than 90° , the intersection angle at point $Q[i+1]$ is then calculated. If this is greater than 90° the interval between i and $i+1$ is the required interval; otherwise the process steps in the direction of increasing i until the correct interval is found. If the magnitude of the intersection angle at the initial point i is greater than 90° , the intersection angle at point $Q[i-1]$ is calculated. If this is less than 90° the interval



between $i-1$ and i is the required interval; otherwise the process steps in the direction of decreasing i , still until the correct interval is found.

Although this process will work regardless of the initial value of i , a well-chosen initial value will improve the efficiency of the process by minimising the number of trials. For a given scan, an initial value for the first tie point pixel can be estimated from the known y coordinate of the sub-satellite point. Thereafter, as the tie point pixels are calculated regularly along the scan, the value of i found for one pixel can be used as the starting point for the next.

Once the correct interval has been found, the sign of the difference $\beta - \varepsilon$ (reduced into the range $-\pi$ to π if necessary), at either end-point, determines on which side of the ground track the pixel P lies, and therefore the sign of the x co-ordinate of P . If $(\beta - \varepsilon)$ is positive, then the point P is to the right (east) of the ground-track and its x co-ordinate will be positive. If $(\beta - \varepsilon)$ is negative, then the point P is to the left (west) of the ground-track and its x co-ordinate will be negative.

Some comments on the actual implementation of this test are appropriate here. Note first of all that the angle β will always be positive (i.e. $0 < \beta < 180^\circ$) because the retrograde ground track always has a westwards component. The angle ε , however, may be negative. If ε is positive, then the difference $(\beta - \varepsilon)$ always lies in $-180^\circ < \beta - \varepsilon < 180^\circ$ and its sign and modulus give the required values. If ε is negative, then $0^\circ < (\beta - \varepsilon) < 360^\circ$. If this is less than 180° it is already in the reduced range and we can return its sign and magnitude as before. If it is greater than 180° the angle should have 360° subtracted first. An equivalent approach is that the required magnitude is $360^\circ - (\beta - \varepsilon)$ and the sign is negative. The table below summarises the situation. The algorithm explicitly tests these four cases.

	$(\beta - \varepsilon) > \pi$	$(\beta - \varepsilon) < \pi$
$\varepsilon < 0$	magnitude = $2\pi + (\varepsilon - \beta)$ sign negative	magnitude = $(\beta - \varepsilon)$ sign positive
	$\beta < \varepsilon$	$\beta > \varepsilon$
$\varepsilon > 0$	magnitude = $(\varepsilon - \beta)$ sign negative	magnitude = $(\beta - \varepsilon)$ sign positive

4.4.4.10.2 Calculation of the sides of the triangle

Once the correct interval Q_1Q_2 has been identified, the co-ordinates X, Y may be calculated. The latitude and longitude of each point P, Q_1 and Q_2 are known, and therefore the length of each side of the triangle PQ_1Q_2 can be determined by means of an inverse formula.

This process is applied in turn to each of the three sides of the triangle PQ_1Q_2 . The triangle is then fully determined and any of its angles can be found. However, this is a triangle on the spheroid, and if we are to base our computations on ellipsoidal geometry, we must use a self-consistent method for solving the triangles. The method adopted is described in the following section.

4.4.4.10.3 Solution of Ellipsoidal triangle.

There is no exact method for treating this case, and so the solution must be based on an approximation. The approximation adopted is based on Legendre's theorem, or more strictly on its extension to the spheroid.

Legendre's Theorem is a result relating to spherical triangles. It is well known, and is not difficult to prove, that the area of a spherical triangle is proportional to the spherical excess of the triangle; this is the amount by which the sum of its three angles exceeds π radians (180°). Thus, suppose ABC is a triangle drawn on a sphere of radius R . The sum of the angles A, B and C will exceed π by the amount

$$E = (A + B + C) - \pi.$$



Then the area of the triangle ABC is ER^2 , where E is in radians. This result is exact.

Let the sides of the triangle ABC be a , b , c and consider the auxiliary plane triangle A'B'C' that has sides of the same length as ABC. The angles of A'B'C' will of course add up to π ;

$$A' + B' + C' = \pi.$$

Legendre's theorem states that to a good approximation for sufficiently small triangles the area of the auxiliary plane triangle A'B'C' is the same as that of the spherical triangle ABC, and that each of its angles can be derived by reducing the corresponding angle of the spherical triangle by one-third of the spherical excess. Thus

$$A' = A - E/3$$

and similarly

$$B' = B - E/3$$

$$C' = C - E/3.$$

It follows that any problem in spherical trigonometry can be reduced to an equivalent problem in plane trigonometry by reducing the angles according to the above equations.

It can also be shown (Bomford [RD-13]) that Legendre's theorem can be extended to triangles on the ellipsoid. Thus the statements above are true (to an appropriate approximation) if the triangle ABC is drawn on the ellipsoid, where that the sides of the triangle are defined as the geodesics joining the vertices A, B, C. In this case the area of the triangle is given by the expression ER^2 as above provided a suitable value for the radius is given. Bomford [RD-13] gives the mean radius

$$R = \left\{ \frac{1}{3} \left(\frac{1}{\rho_1 \nu_1} + \frac{1}{\rho_2 \nu_2} + \frac{1}{\rho_3 \nu_3} \right) \right\}^{-1/2},$$

in which the subscripts 1, 2 and 3 refer to the three vertices A, B and C, and also gives higher order terms for this case. There are two consequences of this.

- (1) Any trigonometric calculation on the spheroid may be reduced to a plane calculation by correcting the angles by $E/3$ as above.
- (2) Because the relationship of the angles of the spheroidal triangle to those of the auxiliary plane triangle are the same as those of the angles of a spherical triangle having the same sides on a sphere of radius R , it follows that to the extent that the higher order terms are negligible, we may use the formulae of spherical trigonometry to solve the spheroidal triangle, if we work with the corrected angles.

4.4.4.10.4 The angle at Q

We now return to the solution of the triangle PQ_1Q_2 . Thus far our calculation has been exact, to the extent that sufficient terms of the expansion in (equation 5.4.46) have been used, given that the lengths so determined will be normal section lengths. The next step is to compute the angle at Q_2 .

The length of the normal section differs from that of the geodesic by a negligible amount for sufficiently short lines, and so we can regard the lengths calculated above as geodesic lengths, and can regard them as the lengths of the auxiliary plane triangle. We can therefore use Legendre's theorem to derive the angles of the geodesic triangle PQ_1Q_2 .

The area of the auxiliary plane triangle is given exactly by a standard formula of plane trigonometry:

$$s = (s_{12} + s_{2p} + s_{p1})/2$$



$$A = \sqrt{s(s - s_{12})(s - s_{2p})(s - s_{p1})}$$

and so the spherical excess is $E = A/R^2$.

By plane trigonometry the angle at Q_2' in the plane triangle is

$$Q_2' = 2 \times \arctan \left\{ \sqrt{\frac{(s - s_{12})(s - s_{2p})}{s(s - s_{p1})}} \right\}$$

hence the angle at Q_2 is this + $E/3$.

We now use the precept (2) above to use the formulae of spherical trigonometry to calculate the x and y co-ordinates. (Strictly this gives us geoid lengths to the point X" at which the geoid through P is orthogonal to the ground track, but the errors are small). It is this final stage that is the most difficult to quantify, but the errors are unlikely to be large in comparison with those due to the assumption that the ground track proper in an interval can be replaced by either the normal section or geodesic between the end-points.

4.4.4.10.5 The x and y co-ordinates

Given the side Q_2P (q_1) and the angle Q_2 , the right-angled triangle PQ_2X (Figure 4-11) is fully determined, and the arcs PX and Q_2X can be calculated. The arc PX is determined by an application of the sine rule.

$$\sin (PX) = \sin (PQ_2) \sin Q_2. \quad (4.4.55)$$

The side Q_2X is determined by an application of the cosine rule to the right-angled triangle.

$$\cos (PQ_2) = \cos (PX) \cos (Q_2X)$$

so

$$\cos (Q_2X) = \cos (PQ_2) / \cos (PX) \quad (4.4.56)$$

The arc lengths ξ , η can be determined by multiplying the angular lengths PX and Q_2X respectively by the radius of the earth:

$$\xi = R (PX) \quad (4.4.57)$$

$$\eta = R (Q_2X) \quad (4.4.58)$$

The sign of the x co-ordinate of P is determined by inspecting the sign of the angle $(\beta - \epsilon)$ derived at the last value of i above. Then the x and y co-ordinates of P, expressed in km, are

$$x = (\text{sign of } x) \xi \quad (4.4.59)$$

$$y = 25i + (25 - \eta) \quad (4.4.60)$$



5 ALGORITHM DESCRIPTION

5.1 LEVEL 0 PROCESSING FOR THE SLSTR INSTRUMENT

5.1.1 The SLSTR Instrument Science Packet

The raw SLSTR instrument data are delivered in binary Instrument Source Packets (ISPs). There are 13 distinct types of ISP,

- Nine optical channel packets (S1 – S9),
- Two fire channel packets (F1, F2),
- One scan packet,
- One housekeeping packet.

All except the housekeeping packet are further differentiated by one of eight target IDs which identify the instrument view and scene from which the packet data is collected, giving a total of 97 distinct ISPs. A total of ten packets are generated for each optical channel and for the scan position in each “cycle”, corresponding to two complete revolutions (“scans”) of the scan mirrors. This is the shortest non-repeating instrument data cycle. One housekeeping packet is generated per scan, or two per cycle.

The ISPs for channels S1 – S9, F1 and F2, and for the scan ISP, may be of variable length, dependent on the pixel map loaded into the instrument processor, up to a well-defined maximum value.

Each ISP will be formatted as a Space Packet following the description given in the CCSPS Space Packet Protocol [RD-8]

The ISP will form the cargo of a SpaceWire packet [RD-8].

The information required to identify the source packet is contained in one of three contiguous headers:

- The Space Packet Primary Header,
- The Packet Utilisation Standard (PUS) Data Field Header,
- An Auxiliary Header which forms part of the packet content.

The primary header contains the PCAT (Table 5-1), a code with value 0 – 15 which identifies the packet type, and the number of packet bytes following the primary header, minus one.



PCAT	ISP Content	Band	Type
0	Channel S1	555nm	0
1	Channel S2	659nm	8
2	Channel S3	865nm	16
3	Channel S4	1.375µm	24
4	Channel S5	1.61µm	32
5	Channel S6	2.25µm	40
6	Channel S7	3.74µm	48
7	Channel S8	11.85µm	56
8	Channel S9	12µm	64
9	Channel F1 (fire)	3.74µm	72
10	Channel F2 (fire)	11.85µm	80
11	Scan position	–	88
12	Housekeeping	–	96

Table 5-1: Assignment of ISP PCAT codes (adapted from [RD-5]). Type is an index value assigned to the PCAT code to allow consistent ISP unpacking in the GPP.

The data field header contains a time stamp derived from the spacecraft clock. This time stamp is updated at cycle acquisitions 0 and 3670 (the first acquisition of each scan) and is identical for all packets associated with the scan.

The definition of the auxiliary header varies between packet types, but the first nine bytes are common to all packet types except for the housekeeping packet.

Content	Length (bytes)
DPM mode	1
Target ID	1
Target first acquisition	2
Target length	2
Validity	1
Scan sync counter	2

Table 5-2: ISP “common” auxiliary header (extracted from [RD-5]).

This “common” auxiliary header (Table 5-2) contains a target ID (Table 5-3) which identifies the instrument view from which the science content was collected. The first scan of each cycle contains the observation sequence D0h, B0h, A1h, C1h, A0h, and the second scan contains the observation sequence B1h, D1h, A1h, C0h, A0h.



Target ID	View	Scene	Type
A0h	Nadir	Swath	0
A1h	Oblique	Swath	1
B0h	Nadir	BB1	2
B1h	Oblique	BB1	3
C0h	Nadir	BB2	4
C1h	Oblique	BB2	5
D0h	Nadir	VISCAL	6
D1h	Oblique	VISCAL	7

Table 5-3: Assignment of ISP target IDs to instrument views (extracted from [RD-6]). Type is an index value assigned to the target ID to allow consistent ISP unpacking in the GPP.

Our understanding of the ISP content is summarised in Figure 5-1 and Table 5-4. The first shows the derivation of the packet information and the associated naming conventions, while the second expands the information into a byte- and bit-ordered table.

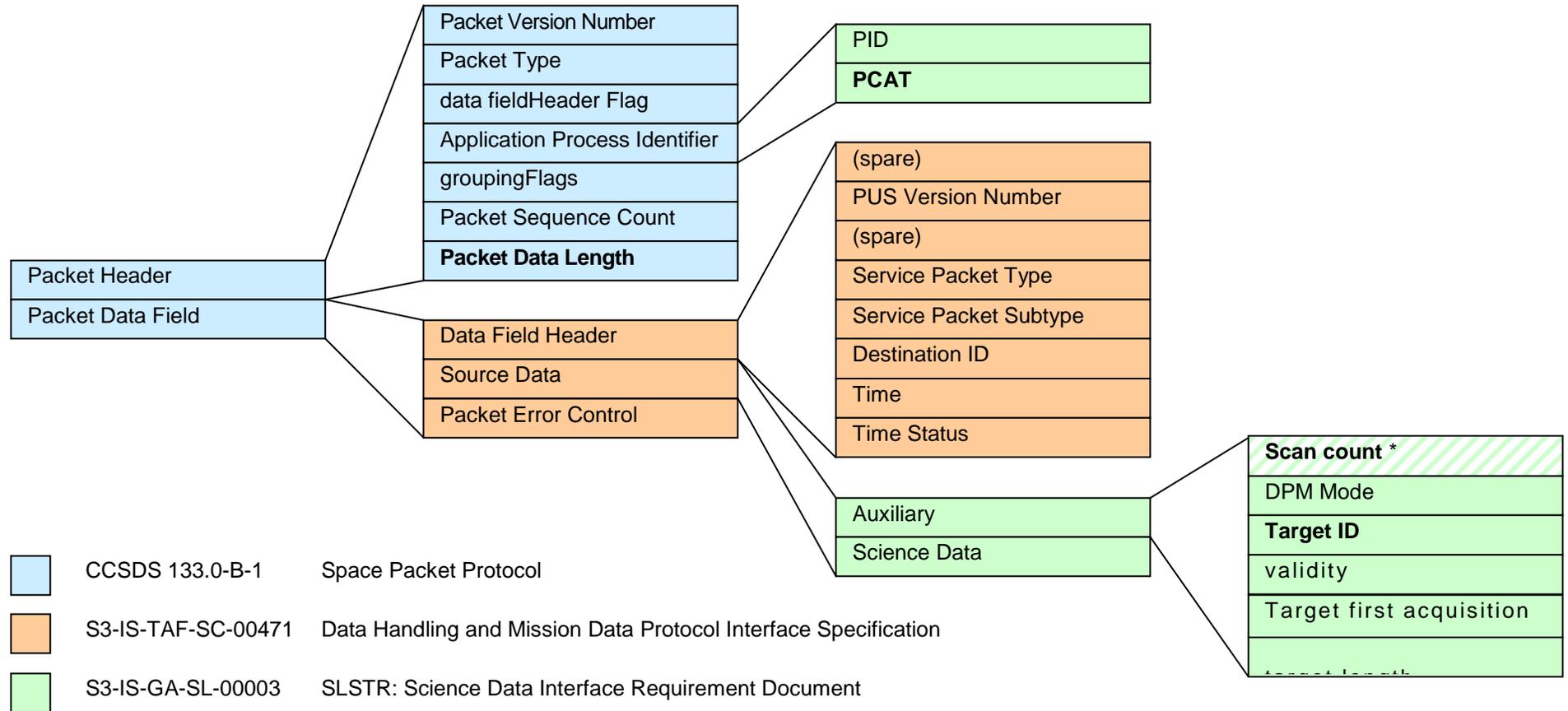


Figure 5-1: The provisional SLSTR Instrument Science Packet (ISP). Sources of information are indicated by colour. Information required to unpack the ISPs are highlighted with bold type. A scan count* is not yet defined in the auxiliary header.



Name	Size (byte.bit)	Value
Packet Version Number	0.3	000b
Packet Type	0.1	0b
Secondary data field Header Flag	0.1	1b
PID	0.7	4Ah
PCAT	0.4	0h – Ch
groupingFlags	0.2	11b
Packet Sequence Count	1.6	
Packet Data Length	2.0	(= n_{PDL})
(spare)	0.1	
PUS Version Number	0.3	001b
(spare)	0.4	
Service Packet Type	1.0	201dec
Service Packet Subtype	1.0	31 dec
Destination ID	1.0	0 dec
Time	7.0	
Time Status	1.0	0dec or 1dec
Scan count	2.0	0-65536
DPM Mode	1.0	0 or 1
Target ID	1.0	00h-FEh
validity	...	0 or 1
Target first acquisition		0-7339
Target length		0-1599
Science Data	$n_{PDL} - 23$	
Packet Error Control	2.0	

Table 5-4: Provisional SLSTR Instrument Science Packet (ISP). For sources of information, refer to Figure 5-1. Information required to unpack the ISPs are highlighted with bold type.

5.1.1.1 Comments

The content of the ISP is provisional. In particular, we have requested a scan counter (not currently defined) to order the packets and to disambiguate those packets which are generated every scan (housekeeping packet, S1 – S9, F1 and F2 nadir and oblique swath packets). We have also suggested that the housekeeping packet should be assigned the “common” ancillary data fields (including at least the scan count and a dummy target ID) so that, for all packet types, a single read can be used to extract all the header information required to unpack the packets.



5.1.2 The SLSTR Level 0 ISP Product

The SLSTR Level 0 ISP product will consist of:

- An ISP-specific XML header,
- Unaltered copies of the binary ISP file or files generated by the instrument or the instrument simulator,
- A searchable XML index to the positions of the ISPs within the binary files.

5.1.2.1 ISP File Unpacking

We presume that the raw binary ISP byte stream(s) will be contained in one or more binary files and that a list (XML or text) of these files will be provided. The following scheme can ingest, unpack and index any configuration of input files (including a single file) without modification.

1. Calculate the size of the unpacked data array:
 - a. Extract the scan number of the first and last scan in the dataset and calculate the number of scans in the unpacked array:

$$n_scan = last.scan_count - first.scan_count + 1$$

Size `n_scan` must be calculated in unsigned integer arithmetic compatible with the definition of the scan counter to cope correctly with counter wrapping.

- b. Assign the number of distinct ISP types:
$$n_type = 97$$
2. Create output arrays of dimension `array[n_scan][n_type]` to hold:
 - a. the ISP headers,
 - b. the ISP content,
 - c. FEP annotations *,
 - d. quality flags,
 - e. file handles,
 - f. file offsets.

3. Initialise all quality flags:

$$quality[[]].packet_absent = 1$$

4. Read in the names of the files containing the ISP data and for each ISP file, assign a unique file handle, `f` (this could just be the index to the file list).

5. For each file, step through the individual ISPs and for each ISP:

- a. Note the file pointer value, `p`
- b. Read the fixed-length file header(s), `h`
- c. Calculate the array indices:

$$i = h.scan_count - first.scan_count$$

$$j = ISP_type(h.PCAT, h.target_ID)$$



Index i must be calculated in unsigned integer arithmetic compatible with the definition of the scan counter to cope correctly with counter wrapping. `ISP_type` returns the sum of the type values in Table 5-1 and Table 5-3.

- d. Calculate the length of the remaining variable-length ISP content:

$$n = h.data_length + 5 - n_header$$

where `n_header` is the size of the fixed length header(s), and read n bytes of ISP content into the content array, `content[i][j][]`

- e. Read the error control value, c

- f. Copy the ISP header(s), file handle and offset to the out put arrays:

```
header[i][j] = h
```

```
handle[i][j] = f
```

```
offset[i][j] = p
```

- g. Perform quality checks * and copy results into the quality flag array:

```
quality[i][j].packet_absent = 0
```

```
quality[i][j].CRC_failed = CRC_check(p, n, c)
```

...

At least the “packet absent” flag must be implemented to enable null entries to be interpreted correctly.

5.1.2.1.1 Comments

No additional annotation information is required to unpack the ISP files. As the L0 ISP content is identical to the raw ISP files, the unpacking scheme can be applied equally to raw ISP or Level 0 ISP files.

It is very unlikely that both TDI and non-TDI SWIR channel ISPs (S4 – S6) will both be present in the data stream, especially as it is now proposed that all TDI processing should be carried out on ground, in which case no TDI SWIR packets will be delivered and PCAT codes 6, 7 and 8 are deprecated. Nonetheless, until this is formally confirmed, it is suggested that one or the other should be specifically selected with a processor switch.

No FEP simulation is envisaged within the GPP perimeter [RD-10]. The L0 product description includes FEP annotations, but we are not aware of the content. We have included a notional annotation array in the unpacking scheme as a placeholder.

We have proposed that the simulator should forward the first and last packet headers as summary information and that in operation the FEP should do the same, but if these are not available, a modified approach will be required. Possibilities include:

1. Make a preliminary pass through the ISP files to identify the first and last packets,
2. Define an oversized “circular” scan array with the maximal number of entries (for a two-byte scan counter, this would be `n_scan = 65536`), and identify the first packet after unpacking,
3. Define a slightly oversized “circular” scan array (for one orbit, `n_scan ~ 20000`), begin filling from an arbitrary point ($i = 0$, say, for the first processed ISP), and identify the first packet after unpacking.

Regardless of method, once the index of the first scan containing data has been identified, all subsequent processes are unchanged (with the exception of some minor indexing arithmetic).



The GPP perimeter document [RD-10] does not require quality checking, on the assumption that the input stream is perfect, but the following ISP checks are performed:

Set all flags for the ISP to 0 (i.e. all OK)

1. CRC check

The CRC check will follow the scheme as defined in Annex A of ECSS--E--70--41A

If error is present then

CRC Error flag for ISP(i) = 1b

2. Consistency of the scan count with the time stamp,

The test to be performed checks that the time interval between successive scans is nominally 301ms as follows.

Note: Care has to be taken to account for the wrap around of the packet sequence counters and instrument scan counters that should occur every 2^{16} scans. If a negative value is detected add 2^{16} to the value before comparing.

$$\Delta t = (\text{time}(\text{scan}_n) - \text{time}(\text{scan}_{n-1})) / (\text{scan_count}(\text{scan}_n) - \text{scan_count}(\text{scan}_{n-1}))$$

IF($\Delta t \neq 301 \pm dt$) then

scan_count_discontinuity flag for scan_n = 1b

Where dt is the resolution of the on board clock

3. Valid fixed header information

If(fixed_header_value(ISP(i)) \neq expected_fixed_header_values) then

Header_Error(ISP(i)) = 1b

4. Valid PCAT codes,

This test checks that the Packet IDs contained in the ISPs fall within the expected range. Values outside this range could indicate a non standard instrument configuration.

If(PCAT (ISP(i)) >Chex) then

Invalid_PCAT(ISP(i)) = 1b

5. Valid target ID codes



This test checks that the target IDs contained in the ISPs fall within the expected range.

According to RD-6, the baseline set of target ID's falls in the range A0h to D1h. Values outside this range could indicate a non standard instrument configuration.

If(Target_ID(ISP(i)) lies outside of range 0Fh to D2h) then

Invalid_Target_ISP(ISP(i)) = 1b

6. Consistent sequence counter values

This test checks that the packet counter is increasing monotonically. A difference between successive packets not equal to 1 could indicate a missing packet or data out of sequence.

Note: As with step 2, care are has to be taken to account for the wrap around of the packet sequence counters and instrument scan counters that should occur every 2^{14} scans. If a negative value is detected add 2^{14} to the value before comparing.

If(seq_count(isp(i))-seq_count (isp(i-1)) \neq 1 then

Sequence_Error(isp(i)) = 1b

7. Duplicate packets.

This test checks that the packet is not a duplicate of one previously processed. The test compares the sequence_count, scan_count and crc_error, time of the ISP against the previous value. If all fields are identical then the packet shall be flagged as duplicate and is not to be used in the processing.

Duplicate_Packet(isp(i)) = 1b

5.1.2.2 ISP Index File

The ISP index file is a binary auxiliary file which contains an ordered and searchable list of references to the positions of the ISPs within the binary ISP files associated with the product and a limited amount of information about the individual ISPs, including <TBC> the ISP size, APID (decoded to a channel), target ID (decoded to a view and scene) and time stamp. The product is described in [RD-11].

The ordering and hierarchy of the ISP entries in the cycle list is essentially the same as that of the ISP entries in the unpacked ISP array, so the contents of the index file can be built up by parsing the ISP array. We presume that the unpacked ISPs are organised in a two-dimensional array, ordered by type and scan, as described in the previous section. For each scan, there are 97 array elements corresponding to the 97 distinct combinations of channel (PCAT) and target (target ID). Reading by type and then by scan, they are ordered by target, then by channel (including scan position and housekeeping), then by scan, then by cycle.

5.1.2.2.1 Writing the ISP Index

The ISP index contains three principle elements: introductory metadata, a list of ISP files and a list of ISPs, including file references and a limited amount of information to allow the list to be searched. An XML entry is opened with a <tag> and closed with a </tag>.



1. Open the index entry.
2. Write any metadata entries (product name, processor version, generation time *etc.*).
3. Write the file list.
4. Write the ISP list.
5. Close the index entry.

5.1.2.2.2 Writing the File List

The purpose of the file list is to associate a file handle with each ISP file URL. This handle will then be used as a shorthand reference in the following ISP list. The only requirement on the handles is that they be unique for each ISP file associated with the product.

1. Open the file list entry.
2. For each file:
 - a. Open the file entry,
 - b. Write the file URL entry,
 - c. Write the file handle entry,
 - d. Close the file entry.
3. Close the file list entry.

5.1.2.2.3 Writing the ISP List

This section describes a parsing scheme to construct the ISP list from the array of unpacked ISPs.

1. Open the ISP list entry.
2. Step through the array elements in the order:

```
for (i = 0; i < n_scan; i++) for (j = 0; j < n_type; j++) {...}
```
3. For each valid element, except for the first:
 - a. Always close the current target entry.
 - b. If the channel has changed from the previous array element, close the current channel entry.
 - c. If the scan has changed from the previous array element, close the current scan entry.
 - d. If the cycle has changed from the previous array element, close the current cycle entry.
4. For each valid element:
 - a. If the cycle has changed from the previous array element:
 - i. Open a new cycle entry,
 - ii. Write the cycle number entry, using the value of `header[i][j].scan_count >> 1`.
 - b. If the scan has changed from the previous element:
 - i. Open a new scan entry,
 - ii. Write the cycle number entry, using the value of `header[i][j].scan_count`,
 - iii. Write the time stamp entry, using the value of `header[i][j].time`.



- c. If the channel has changed from the previous array element:
 - i. Open a new channel entry,
 - ii. Write the channel name entry, using one of <TBC>: "S1", "S2", "S3", "S4", "S5", "S6", "S7", "S8", "S9", "F1", "F2", "scan" or "AUX", dependent on the value of `header[i][j].PCAT` (Table 5-1).
- d. Always:
 - i. Open a new target entry,
 - ii. If not a housekeeping packet, write the instrument view, using one of: "nadir" or "oblique", dependent on the value of `header[i][j].target_ID` (Table 5-3),
 - iii. If not a housekeeping packet, write the instrument scene, using one of: "swath", "BB1", "BB2" or "VISCAL", dependent on the value of `header[i][j].target_ID` (Table 5-3),
 - iv. Write the file handle entry, using the value of `handle[i][j]`,
 - v. Write the ISP offset entry, using the value of `offset[i][j]`,
 - vi. Write the ISP size entry, using the value of `header[i][j].data_length + 7`.
5. After the last entry has been written, and if at least one entry has been written:
 - a. Close the current target entry
 - b. Close the current channel entry,
 - c. Close the current scan entry,
 - d. Close the current cycle entry.
6. Close the ISP list entry.

5.1.2.2.4 Comments

Most metadata will be placed in separate general and ISP-specific XML headers [RD-10]. Only minimal metadata information is envisaged for the XML index file.

It is in principle possible to extract additional information from the ISPs and the unpacking process and to include it as additional entries in each target entry. Possible candidates include the quality flags and FEP annotations, if available. However, it is assumed here that only the information required to distinguish and extract the different "flavours" of ISP (PCAT/target ID) is needed and that any further information will be taken directly from the ISPs themselves.

The beauty of the XML schema is that additional information can be introduced if required at a later stage with minimal modification to the existing XML writing process or to any reading tools, as the index is not format-dependent. It also copes straightforwardly with missing ISPs.



5.1.3 NAVATT File Processing

The structure of the NAVATT packets is described in RD-14. Assuming the simulator output includes a file containing NAVATT ISPs, the L0 processing scheme described in section 5.1.2 can be employed. Also refer to 5.4.3.2.1.



5.2 OVERVIEW OF LEVEL 1B PROCESSING FOR THE SLSTR INSTRUMENT.

The following description of the expected SLSTR data processing is based on that used for the AATSR instrument processing facility, suitably modified. This is appropriate since the basic functionality of the SLSTR instrument is derived from that of AATSR, and so the necessary processing steps are largely unchanged. Moreover the algorithms are well-established and provide a robust baseline for the operational applications of the SLSTR instrument. The detailed algorithms will require amendment and possible enhancement to accommodate the higher ground sampling and wider swath widths of the SLSTR instrument.

Figure 5-2 and Figure 5-3 show an overview of the Level 1b processing.

The SLSTR Level 1 processor will operate on segments of Level 0 data of approximately one orbit. In the case of NRT data this may represent the data between downlinks to the ground station; for consolidated data it will be a segment of data slightly longer than the interval between consecutive ascending nodes. The data stream must exceed a single orbit because of the displacement of the nadir and oblique views and the curvature of the instrument scans; for example, if the oblique view is directed backwards, the oblique view at the end of the orbit is measured after the sub-satellite point has passed the ascending node.

The Level 0 data product will comprise a series of records presented in chronological sequence, each containing a single instrument source packet. It is assumed that each source packet represents the two scans (nadir and oblique) of a synchronous pair. With the exception of the determination of calibration parameters, which will be derived from the average across a range of scans, the data processing up to but not including the re-gridding stage will treat each source packet independently. For the visible channel calibration, the channel gains will be determined once per orbit by averaging over the block of scans for which the VISCAL target is visible.

In the following we give a brief description of each major processing stage, and an indication of the requirement for auxiliary data by that step.



5.2.1 Level 1a Processing

Figure 5-2 summarises these processing steps.

Source Packet Processing (Stages 1 - 6)

The purpose of these steps is to unpack and validate the source packet data and convert it to engineering units for the higher processing levels that generate the products and geolocate the data. The main functions of these steps can be summarised as follows:

1. **Source Packet Quality Checks:** Performs basic quality checks on each raw packet, ensuring that only those that pass the checks contribute to the product.
2. **Unpack Ancillary Data:** Unpacks all required ancillary and housekeeping data including the temperatures of the on-board black bodies and instrument health and status information.
3. **Validate Unpacked Ancillary Data:** validates the unpacked ancillary and housekeeping data.
4. **Convert Unpacked Ancillary Data:** Converts to engineering units those items of ancillary data, including the temperatures of the on-board black bodies and other instrument temperatures and data items, for which a conversion to engineering units is defined.
5. **Validate Converted Ancillary Data:** Validates those converted ancillary data items that are essential to the calibration.
6. **Science Data Processing:** Unpacks and validates the science data containing the earth view and black body pixel counts for all available channels from each packet.

The inputs, outputs, and relationship between each Stage are summarised in Figure 5-2

The following auxiliary input data sets are required by these processing stages:

- **Level 1B Processor Configuration File:** defines processor configuration data, internal error codes and other items.
- **SLSTR Instrument Data File:** defines source packet data structures, validation parameters and engineering unit conversions for the SLSTR instrument.

Infra-Red Channel calibration (Stage 7)

7. **Infra-red Channel Calibration:** This step calculates the calibration offset and slope that describe the linear relationship between pixel count and radiance for the thermal IR channels. The parameters are determined from the black body pixels counts and the black body temperatures, and the process makes use of look-up tables for the conversion of temperature to radiance.

The following auxiliary input data set is required by this processing stage:

- **SLSTR Calibration Data File:** defines temperature to radiance look-up tables and non-linearity correction tables for the SWIR channels where required.

Solar Channel calibration (Stage 8)

8. **Solar Channel Calibration:** This step unpacks the VISCAL data once per orbit (if present), when it is detected that the VISCAL unit is in sunlight, and calculates calibration parameters for all visible channels. These calibration parameters may not be used to calibrate the science data in the visible channels for the current orbit (which may make use of VISCAL data from a previous orbit), but are written to the Visible Calibration Coefficients ADS in the GBTR product.



The following auxiliary input data set is required by this processing stage:

- Visible Calibration Data File: the Visible Calibration Coefficients ADS from a previous orbit.

Satellite Time Calibration (Stage 9)

9. Satellite Time Calibration: The OBT associated with each the SCANSYNC pulse of each source packet are converted to UTC. The time of each pixel is also calculated.

The following auxiliary input data set is required by this processing stage:

- Satellite time calibration data.TBC

Geolocation

Stages 10 to 14 relate to geolocation and related matters.

10. Generate Geolocation grid: This stage, which is called only once in the processing of an orbit, generates look-up tables for use in later geolocation and image pixel co-ordinate determination stages. The tables comprise: (i) tables of the latitude, longitude, and y co-ordinate of a series of sub-satellite points that define the satellite ground track, for use in the calculation of the pixel x and y co-ordinates; (ii) tables of the latitudes and longitudes of a rectangular grid of tie points covering the satellite swath.
11. Determine Sentinel 3 Orbit: This Stage uses the CFI with the NAVATT to determine the position and orientation of the platform at the time of each scan.
12. Geolocate Pixels: Orhto-geolocation is performed on selected tie point pixels of each scan. The direction of the line of sight to each tie point pixel is calculated, and used to determine the geodetic latitude and longitude, on the reference ellipsoid, of the tie point.
13. Calculate Pixel x and y co-ordinates: The x-y (across-track and along-track) coordinates of each tie point pixel are derived from the pixel latitude and longitude.
14. Calculate Solar and Viewing Angles: Solar and viewing angles required for cloud clearing are also determined at this stage. The angles are calculated using the CFI at a series of tie points around the scan at increments of 50 km (TBC) in x.

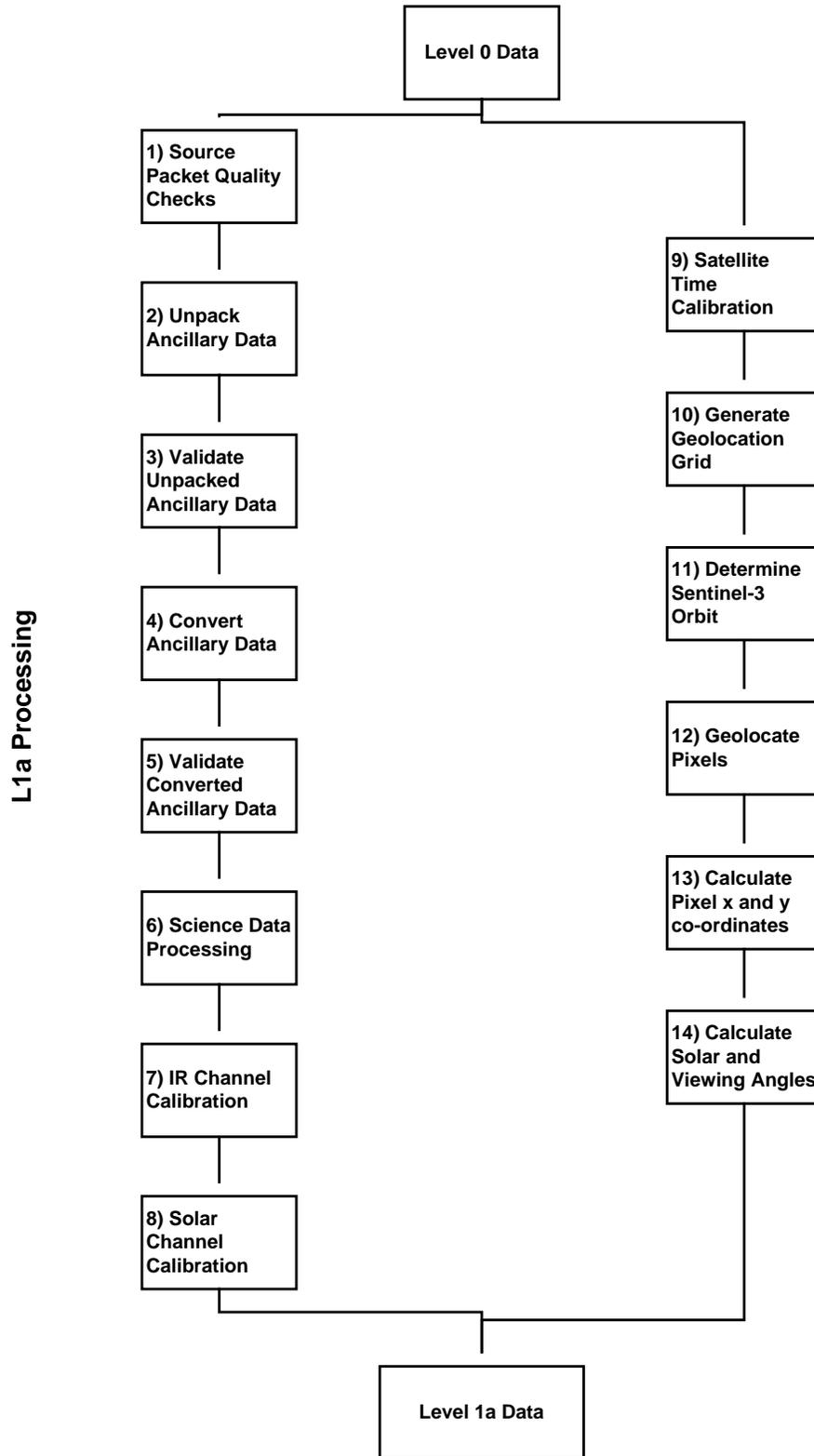


Figure 5-2: Level 1a Processing



5.2.2 Level 1b Processing

Figure 5-3 summarises these processing steps.

Signal Calibration (Stage 15)

15. Uncalibrated scan pixels are calibrated using calibration coefficients derived earlier. Pixel calibration uses these calibration coefficients to convert the pixel data to brightness temperature, in the case of the infra-red channels, or to radiance in the case of the visible and near-visible channels.

The following auxiliary input data set is required by this processing stage:

- SLSTR Calibration Data File: defines radiance to temperature look-up tables and non-linearity correction tables for the SWIR channels where required.

Time Domain Averaging (Stage 16)

16. In order to provide resolutions of 500 m in the SWIR channels, the design of the current FPA for these channels incorporates a 2 by 4 array of detectors, the long edge being aligned in the along-track direction. As a consequence, the swath is effectively scanned twice in the across-track direction, once by each column of the detector array, and so it is appropriate to perform averaging on these duplicate samples to reduce noise. Note that at the swath edge the pixels of the resultant averaged image will be distorted, but the individual plines will be retained to support level 1c processing.

Note that this step follows signal calibration so that the averaging is applied to calibrated reflectances. This is because the calibrations of the detectors in the two columns may differ.

Regrid Pixels (Stage 17)

17. Calibrated SLSTR pixels are regridded into co-located oblique and nadir images, onto a 1 km grid using the pixel positions derived previously. The process used in (A)ATSR processing has migrated instrument pixels to the nearest grid point. This is an input-driven process, in the sense that the process loops over input pixels, and is a highly efficient process in computational terms, but it has been suggested that an output-driven process (looping over image pixels) would present advantages. If such an algorithm were adopted, the next process (cosmetic fill) would not be required.

Cosmetic Fill (Stage 18)

18. Cosmetic filling of nadir/oblique view images is performed, to fill missing image pixels.

Image Pixel Positions (Stage 19)

19. Linear interpolation is performed to determine the latitude and longitude coordinates of the grid pixels for use in the subsequent stages.

Determine Land-Sea Flag (Stage 20)

20. Given the image pixel latitude and longitude, the surface type for each image pixel is derived using the land/sea flagging algorithm.

The following auxiliary input data set is required by this processing stage:

- Land-Sea Mask Data File.



Cloud clearing (Stage 21)

21. The cloud-clearing algorithms are used to identify image pixels as cloudy or cloud-free. Baseline AATSR processing uses nine independent tests based on the AATSR suite of channels, and has recently been expanded by the addition of a new visible channel test. The additional channels proposed for SLSTR will permit additional tests yet to be specified.

The following auxiliary input data sets are required by this processing stage:

- Land-Sea Mask Data File.
- Cloud LUT Data File: contains look-up tables used by the cloud clearing tests.

Meteo annotations (stage 22)

This processing step aims to provide annotation data at a subset of the product pixels, referred to as the tie points grid. These annotations mainly concern acquisition geometry and meteorological data that are required by Level 2 processing.

The subset of pixels is still TBD, as it shall be adapted to the output product grid – still under discussion – in order to allow easy interpolation at pixel level.

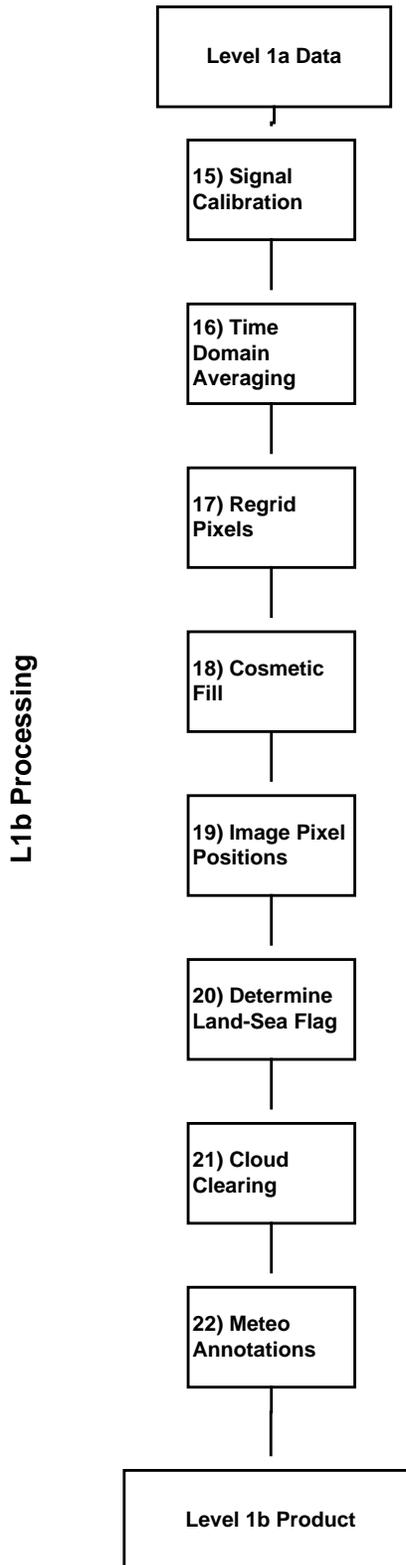


Figure 5-3: Level 1b Processing



5.3 GEO-REFERENCING

The processing modules described in this section, which are a sub-set of those classified as Level 1a processing algorithms, are concerned with aspects of geolocation.

Geo-referencing or geolocation determines the location on Earth of the instrument samples. Taking the satellite orbit data as given, the geolocation of a given pixel depends only on the scan position of the pixel and the measurement time, so that the geolocation is largely independent of the other processing, and can be applied as soon as these quantities have been decoded.

There are 4 basic geolocation grids for each view

- 1km for TIR channels
- 0.5km for VIS and 'A' stripe SWIR channels
- 0.5km for 'B' stripe SWIR channels
- 0.5km for TDI SWIR, which is the average of the 'A' and 'B' stripes

The geolocation is performed for each PIXSYNC and each detector number separately, and then combined for each channel group as above.

5.3.1 Algorithm Input

The following auxiliary input data sets are required by these processing stages:

- Satellite time correlation file, to convert satellite binary time to UTC;
- Platform navigation and attitude data: An orbit state vector or restituted orbit file;

(Note that the above may be derived from the NAVATT data.)

- The scan time and scan position of the instrument samples pointing vectors;
- Earth surface model (ellipsoid + DEM).
- Detector position offsets in the focal plane.

In the SLSTR instrument, the FPA of each channel includes multiple detector elements, each of which is sampled 3670 (TBC) times per scan (although only those samples that correspond to genuine ground or calibration signals are retained in the telemetry). There are two such elements in the long-wavelength channels, and eight in the short-wave and visible channels. Each such element represents a single pixel. It is therefore necessary to account for their relative positions in the focal plane, and this is the purpose of the position offset data set.

Strictly, the arrays of detector elements may not be precisely aligned between the channels. We assume that the mechanical alignment is sufficiently good that the same set of position elements will apply to each channel group (2 element (TIR), 4 element (VIS) or 8 element (SWIR)) so that it is not necessary to specify different constants for each channel. Where more precise alignment is sought in the generation of the level 1c vegetation product, it is assumed that this is provided by the relative shift estimation of the Level 1c algorithm.

5.3.2 Processing Objective

The objective is to determine the co-ordinates of each instrument pixel. The positions of SLSTR pixels will be determined in two co-ordinate systems:



- The geodetic latitude and longitude will be determined with respect to the WGS 84 reference ellipsoid;
- The x and y co-ordinates of the pixel will be determined with respect to the quasi-Cartesian system described in Section 4.4.3. These co-ordinates are required to facilitate the regridding process by which the two views are co-located onto a common grid.

The stages of the calculation (Figure 5-4) have already been summarised in Section 5.2.1. The descriptions are repeated, with some additional detail, in the following.

5.3.2.1 Satellite Time Calibration (Stage 9)

The times associated with each instrument scan are converted to UTC. The objective of this stage is to provide a single UTC value associated with the scan for use as the argument of the ephemeris.

5.3.2.2 Geolocation (Stage 10-14)

Stages 10 to 14 (Figure 5-4) relate to ortho-geolocation and related matters.

10. Generate Geolocation grid: This stage, which is called only once in the processing of an orbit, generates look-up tables for use in later geolocation and image pixel co-ordinate determination stages. The tables comprise: (i) tables of the latitude, longitude, and y co-ordinate of a series of sub-satellite points that define the satellite ground track, for use in the calculation of the pixel x and y co-ordinates; (ii) tables of the latitudes and longitudes of a rectangular grid of tie points covering the satellite swath.
11. Determine Sentinel 3 Orbit: This Stage uses the CFI with the NAVATT to determine the position and orientation of the platform at the time of each pixel of every scan.
12. Geolocate Pixels: Orthogeolocation is performed on each pixel of each scan.
13. Calculate Pixel x and y co-ordinates: The x - y (across-track and along-track) coordinates of each pixel are derived from the pixel latitude and longitude.
14. Calculate Solar and Viewing Angles: Solar and viewing angles required for cloud clearing are also determined at this stage. The angles are calculated using the CFI at every pixel and then sub-sampled at increments of 50 km (TBC) in x .

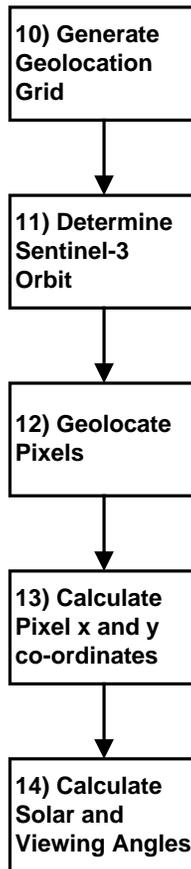


Figure 5-4: Geolocation Modules

5.3.3 Mathematical Description

5.3.3.1 Notation

Geolocation is applied to each measured instrument pixel. Pixels will be identified by a combination of scan number, S , pixel number p , and detector element index k (cf. Section 4.2.3).

The SLSTR will include two independent scan mirrors to generate the nadir and oblique scans. These will be synchronous, so that there is a one-to-one correspondence between pairs of scans, and a single index will suffice to identify them. Each thermal infra-red detector element will be sampled $N_SAMPLE = 3670$ (TBC) times per scan; reflectance channel detector elements will be sampled at twice this rate.

The Level 0 data will comprise instrument source packets each of which will contain instrument data measured during a single scan. Logically it would be possible to have all instrument data from a single pair of nadir and oblique scans in a single telemetry source packet. In practice the sentinel design will not necessarily implement this option, and data from different channels may appear in different source packets (e.g. data from the visible and thermal channels may be placed in different packets. However, it should be possible to assemble packets corresponding to the same scan in ground processing, and so we define a logical source packet to be the aggregate of all data from a single pair of simultaneous instrument scans.



We assume that the channels are sampled synchronously, and that within each channel the detector elements $k = 1, \dots, k_{max}$ are sampled simultaneously. The absolute pixel number p will index the samples of a single detector element the scan. We will need to index the nadir and along-track samples independently, and will distinguish them by the suffixes n and a respectively. Then we have:

S	Scan number;
p_n	Absolute pixel number, nadir scan ($1 \leq p_n \leq N_SAMPLE$)
p_a	Absolute pixel number, oblique (inclined) scan ($1 \leq p_a \leq N_SAMPLE$)
k	Detector element index. $k = 0, 1$ for the thermal channels, $k=0, 4$ for the visible channels, $k=0, \dots, 7$ for the SWIR channels.
t	acquisition cycle (relevant for 0.5km channels), $t=0, 1$

The telemetry does not contain every sample for a given scan; only those pixel samples that correspond to valid data are downlinked. Thus the downlinked pixels will comprise blocks corresponding to the black body, Earth view and VISCAL sections of the scan, selected according to a pixel selection map. We define relative pixel numbers as follows:

$$p'_n = p_n - FIRST_NADIR_PIXEL_NUMBER \quad \text{Relative pixel number, nadir view.}$$

$$p'_a = p_a - FIRST_OBLIQUE_PIXEL_NUMBER \quad \text{Relative pixel number, oblique view.}$$

These quantities index the earth view pixels relative to the first such pixel in the scan; `FIRST_NADIR_PIXEL_NUMBER` and `FIRST_OBLIQUE_PIXEL_NUMBER` are the absolute pixel numbers of the first pixels in the nadir and oblique scans respectively.

5.3.3.2 Common Procedures

This section describes certain procedures which are used by subsequent processing on more than one occasion, and which can therefore be implemented as independent subroutines or procedures. They are presented separately here to simplify the presentation and to indicate how the algorithm may be made more modular. Three such procedures are described:

- solve spherical triangle
- Normal section length and azimuth
- Normal section endpoint

The procedures following make use of the `atan2` function; this represents the arc tangent function of two arguments defined in the conventional way, `atan2(y, x)` being the angle whose tangent is (y/x) and whose quadrant is defined by the signs of x and y . Thus `atan2(sin θ , cos θ)` would return θ .

5.3.3.2.1 Solve spherical triangle

Purpose: to solve for the unknown side and angles of a spherical triangle, given two sides and an included angle. The basis of the method is described in Section 4.4.4.9.4.)

Notation: Suppose ABC is a spherical triangle. We denote by A, B, C the angles at the corresponding vertices, and according to the usual convention we denote by a, b and c the arc lengthsm, in radians, of the sides opposite the vertices A, B, C respectively. Thus a represents the side BC, and so on.

Input parameters: sides b, c and included angle A. (Units of radians)

Output parameters: the side a and the remaining angles B, C. (Units of radians)

Processing steps (See section 4.4.4.9.4 for the derivation of these equations):

$$\cos a = \cos b \cos c + \sin b \sin c \cos A$$

$$B = \text{atan2}\{\sin b \sin c \sin A, (\cos b - \cos a \cos c)\}$$



$$C = \text{atan2}\{\sin b \sin c \sin A, (\cos c - \cos a \cos b)\}$$

$$a = \text{atan2}\{(\sin b \sin A / \sin B), (\cos b \cos c + \sin b \sin c \cos A)\}$$

We denote this transformation by

Solve spherical triangle $\{A, b, c; B, C, a\}$

Here and in the following subsections we adopt the convention that the input parameters of the procedure call precede the semicolon, while the output parameters follow it.

5.3.3.2.2 Normal section length and azimuth

Purpose: given the geodetic latitude and longitude of two points P_1 and P_2 on the ellipsoid, to find the length and azimuth of the normal section through P_2 at P_1 . The method is an implementation of Robbins's formula, described in Section 4.4.4.9.2

Input parameters:

φ_1	geodetic latitude of the point P_1	radians
λ_1	longitude of the point P_1	radians
φ_2	geodetic latitude of the point P_2	radians
λ_2	longitude of the point P_2	radians

Output parameters:

L	length of the normal section at P_1 through P_2	km
β	azimuth of the normal section at P_1 through P_2	radians

Calculate the radii of curvature in prime vertical at the points P_1, P_2 as follows:

$$v_1 = \text{EARTH_MAJOR_AXIS} / (1 - e^2 \sin^2(\varphi_1))^{3/2} \quad \text{eq 5.3-1}$$

$$v_2 = \text{EARTH_MAJOR_AXIS} / (1 - e^2 \sin^2(\varphi_2))^{3/2} \quad \text{eq 5.3-2}$$

$$S_\psi = (1 - e^2) \sin \varphi_2 + (v_1/v_2) e^2 \sin \varphi_1 \quad \text{eq 5.3-3}$$

$$C_\psi = \cos \varphi_2 \quad \text{eq 5.3-4}$$

$$\psi = \text{atan2}(S_\psi, C_\psi) \quad \text{eq 5.3-5}$$

(Note e is the eccentricity of the ellipsoid) Define the angles (expressed in radians)

$$b = \pi/2 - \varphi_1 \quad \text{eq 5.3-6}$$

$$c = \pi/2 - \psi \quad \text{eq 5.3-7}$$

$$\alpha_{12} = \lambda_1 - \lambda_2.$$

It may be convenient to modify α_{12} , by adding or subtracting 2π , to bring it into the range $-180^\circ \leq \alpha_p < 180^\circ$. This case arises if the 180° meridian intersects the arc P_1P_2 .

Solve the following spherical triangle by the method of section 4.8.3.6.

$$\text{Solve spherical triangle } \{A \leftarrow \alpha_{12}, b \leftarrow \psi_1, c \leftarrow \psi_2; \gamma, \beta, \sigma_{12}\}$$

The angle β represents the azimuth of P_2 as measured at P_1 . If only this azimuth is required, the following steps can be omitted. Otherwise correct the arc length to spheroidal length. Calculate



$$g = \varepsilon \sin \varphi_1 \cos \varphi_1 \cos \beta$$

$$h = \varepsilon \cos^2 \varphi_1 \cos^2 \beta$$

Where $\varepsilon = e^2 / (1 - e^2)$

Calculate the line segment between the points:

$$s_{12} = v_1 \sigma_{12} \left[1 - \frac{\sigma_{12}^2}{6} h(1-h) + \frac{\sigma_{12}^3}{8} g(1-2h) \right]$$

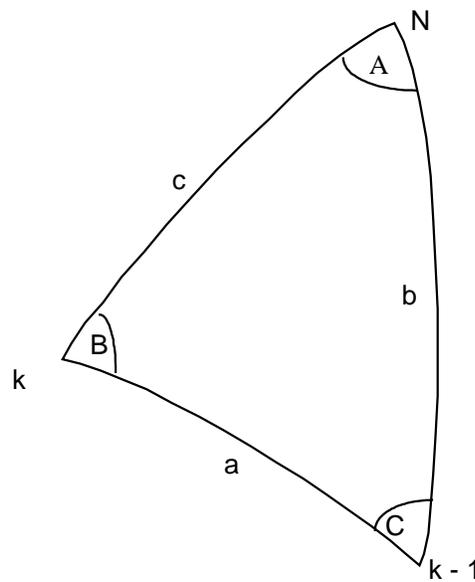


Figure 5-5

We denote this procedure by

Normal section length $\{\varphi_1, \lambda_1, \varphi_2, \lambda_2; L, \beta\}$

5.3.3.2.3 Normal section endpoint

Purpose: given the geodetic latitude and longitude of a point P_1 on the ellipsoid, to find the geodetic latitude and longitude of the point P_2 at a given distance from P_1 on the normal section through P_1 of azimuth β . The method is an implementation of Clarke's direct formula, described in Section 4.4.4.9.3.

Input parameters:

φ_1	geodetic latitude of the point P_1	radians
λ_1	longitude of the point P_1	radians
L	length of the normal section at P_1 through P_2	km
β	azimuth of the normal section at P_1 through P_2	radians

Output parameters:



φ_2	geodetic latitude of the point P_2	radians
λ_2	longitude of the point P_2	radians

Processing steps:

Calculate

$$r_2 = -\varepsilon \cos^2 \varphi_1 \cos^2 \beta \quad \text{eq 5.3-8}$$

$$r_3 = 3\varepsilon(1 - r_2) \cos \varphi_1 \sin \varphi_1 \cos \beta \quad \text{eq 5.3-9}$$

where ε is as defined above. Calculate the radius in prime vertical at latitude φ ,

$$v_1 = EARTH_MAJOR_AXIS / (1 - e^2 \sin^2 \varphi_1)^{1/2} \quad \text{eq 5.3-10}$$

$$g = \frac{L}{v_1} \left[1 - \frac{r_2(1+r_2)}{6} \left(\frac{L}{v_1} \right)^2 - \frac{r_3(1+3r_2)}{24} \left(\frac{L}{v_1} \right)^3 \right] \quad \text{eq 5.3-11}$$

$$\psi_1 = \frac{1}{2} \pi - \varphi_1 \quad \text{eq 5.3-12}$$

The quantities α and ψ_2 are given by the solution of a further spherical triangle:

$$\text{Solve spherical triangle } \{\beta, \psi_1, g; \delta, \alpha, \psi_2\} \quad \text{eq 5.3-13}$$

$$\rho = 1 - \frac{1}{2} r_2 g^2 - \frac{1}{6} r_3 g^3 \quad \text{eq 5.3-14}$$

$$S_j = \cos \psi_j = e^2 \rho \sin \varphi_1 \quad \text{eq 5.3-15}$$

$$C_j = (1 - e^2) \sin \psi_j \quad \text{eq 5.3-16}$$

$$\varphi_2 = \text{atan2}(S_j, C_j) \quad \text{eq 5.3-17}$$

$$\lambda_2 = \lambda_1 + \alpha \quad \text{eq 5.3-18}$$

We denote this procedure by

$$\text{Normal section end point}\{\varphi_1, \lambda_1, L, \beta; \varphi_2, \lambda_2\}$$

5.3.3.3 Satellite Time Calibration

This step is performed for every ISP. The time is computed on four grids, the TIR grid, stripe A, stripe B and TDI grid where the TDI time is simply the average of the A and B grid times.

5.3.3.3.1 Convert scan time to UTC

The OBT given in the ISP header is converted to UTC using the information in the NAVATT packets.

5.3.3.3.2 Pixel time calibration

The time given in the ISP header is the time at the beginning of the scan cycle relating to the data contained in that ISP. This is denoted here as $t_{\text{scan_start}}$. The time of every pixel acquisition is calculated using the assumption of a constant scan speed. Due to differences in the detectors of some channels, the pixel time is not the same for all channels.



The time t_{pix_10} of every pixel within a scan cycle for the 1km resolution channels are calculated using the following algorithm.

$$t_{pix_10} = t_{scan_start} + (target_position * PIX10SYNC) + offset$$

where for all channels except S7F (F1),

$$offset = PIX10SYNC / 2 \text{ in units equivalent to } t_{scan_start}$$

$PIX10SYNC = 81.7\mu s$, the period of the 1km resolution channel pixel acquisitions.

$target_position$ of the n th pixel readout in each ISP is determined using the target first acquisition in the ISP header. $target_position = target_first_acquisition + n$ (where $0 < n < target_length - 1$)

Note that the offset is required because the exact time at which a pixel is measured is not at the beginning of a PIXSYNC period, but the centre of the cycle.

The FIRE channel on detector S7 differs from the other channels. The F2 channel uses the S8 detectors, but the F1 channel uses detector elements that are adjacent to the S7 detector and so the data which are acquired in the same PIX10SYNC period from the two fire channels are for different ground samples. The information relevant for the same ground sample will be at different times from these channels. F2 will have pixel times identical to the other channels as it shares the S8 detectors, but F1 pixel times will differ. In the oblique view, the acquisition from F1 taken at time i has to be compared with the acquisition from F2 taken in time $i+1$. In the nadir view the reverse statement is true. Therefore, the offset used for channel F1 must take account of this.

For S7F (F1),

$$offset = PIX10SYNC/2 + PIX10SYNC \text{ for the oblique view and}$$

$$offset = PIX10SYNC/2 - PIX10SYNC \text{ for the nadir view}$$

The time t_{pix_05} of every pixel within a scan cycle for the 0.5km resolution channels are calculated in a similar fashion using the algorithm outlined below.

$$t_{pix_05} = t_{scan_start} + (2 * target_position + t) * PIX10SYNC/2 + offset$$

where for all channels except stripe B of the SWIR channels (S4-S6)

$$offset = PIX10SYNC / 4$$

t denotes the acquisition cycle (0 or 1)

The B stripe detectors of the SWIR channels need the offset to be shifted by ± 1 pixel, depending on the view.

For SWIR stripe B channels,

$$offset = PIX10SYNC/4 + PIX10SYNC/2 \text{ for the nadir view and}$$

$$offset = PIX10SYNC/4 - PIX10SYNC/2 \text{ for the oblique view}$$

The TDI time grid is calculated as

$$t_{pix_TDI} = (t_{pix_05A} + t_{pix_05_B}) / 2$$



In summary, there are therefore 5 different sets of pixel timings for SLSTR:

t_pix_10 pixel time associated with TIR channels S7-S9 and F2

t_pix_10F1 pixel time associated with F1

t_pix_05A pixel time associated with VIS channels S1-S3 and A stripe of SWIR channels S4-S6

t_pix_05B pixel time associated with B stripe of SWIR channels S4-S6

t_pix_TDI pixel time associated with TDI of SWIR channels

However the F1 channel is assigned the t_pix_10 pixel timings as the difference is negligible

5.3.3.4 Generate Geolocation Grid

This section defines the look-up tables for use in later geolocation and image pixel co-ordinate calculations.

Notation

T_0 The start time of the output (Level 1b) product.

ΔT Along track sampling interval

ak Along-track index

$s(ak)$ Along-track distance of sample ak . The calculation of this section is independent of the instrument geometry.

K is a constant offset that defines the number of samples prior to the ascending node prior to the start of the table, or output product start time if different. K =additional_scan_constant in the L1b processor ADF.

Step 5.3.3.4.1 Generate along-track time-distance LUT

A look-up table defining points on the ground track at time interval ΔT is generated using the NAVATT data. The time step ΔT is chosen to correspond to one granule and calculated as follows:

$$\Delta T = \text{cycles_per_tie_point} * \text{scans_per_cycle} * (\text{cycle_period} / 2.0).$$

The sequence of times $t(k)$ is defined by

$$t(ak) = T_0 + (ak - K)\Delta T, \quad ak = 0, 1, 2, 3, \dots, \quad \text{eq 5.3-19}$$

The time T_0 represents the start time of the output product, which in the case of a consolidated orbit product we expect to be the ascending node of the orbit. Because the leading edge of the nadir scan precedes the sub-satellite point, instrument scans measured before T_0 are necessary to fill in the image at the start of the orbit, and the origin of the look-up tables must precede T_0 by the interval $K\Delta T$, where K is an auxiliary constant whose value is to be found in the additional_scan_constant in the L1b processor ADF.

An initialisation call to the CFI function xo_orbit_file_precise is required at this point, to initialise the orbit calculations.

status = xo_orbit_init_file_precise(input parameters, output parameters)

Input Parameters:

&sat_id	ID must be previously initialised.
&model_id	ID to be previously initialised; astronomical models etc
&time_id	ID to store time correlations. Should have been initialised as part of time correlation function



&orbit_file_mode	Flag to indicate type of orbit file(s) supplied
&n_files	Number of input files
input_files	String array of (I presume) input file names
&time_mode	Flag to indicate how time range is specified
&time_ref	Time reference ID
&time0	Start time of orbit calculation (if applicable)
&time1	End time of orbit calculation (if applicable)
&orbit0	Abs. orbit number of start orbit (if applicable)
&orbit1	Abs. orbit number of end orbit (if applicable)
&precise_conf	Configuration parameters if precise orbit used

Output parameters

&val_time0	Start of initialization validity range
&val_time1	Start of initialization validity range
&orbit_id	This is the output ID.
ierr	Status error vector

In addition to initialising the precursor IDs, we need to identify the correct orbit data to supply to this call. This should be in the same form as is defined for OLCI.

The CFI function `xo_osv_compute` is then used to find the latitude and longitude of the sub-satellite point at time $t(ak)$ for each ak .

```
status = xo_osv_compute (input parameters, output parameters);
```

Input Parameters

(&orbit_id	From the initialisation above
&mode	Propagation model – dummy for current version
&time_ref	As above
&time	Reference time

Output parameters

pos_out	Osculating position vector at pixel time (Earth fixed CS)
vel_out	Osculating velocity vector at pixel time (Earth fixed CS)
acc_out	Osculating acceleration vector at pixel time (Earth fixed CS)
ierr	Status error vector

The output parameters generated by this call are not those required at this point. The required parameters are the latitude and longitude of the sub-satellite point, and the ground trace velocity components. These require a call to `xo_osv_compute_extra`

```
status = xo_osv_compute_extra(input parameters, output parameters);
```

Input Parameters

&orbit_id	From the initialisation above
&extra_choice	Flag to allow an ancillary results choice

Output parameters;

model_out	Vector of model-dependent parameters
-----------	--------------------------------------



extra_out	Vector of model-independent parameters; depends upon extra choice
ierr	Status error vector

The required outputs are the latitude and longitude of the subsatellite point at $t(ak)$:

$$\text{track_long}[ak], \text{track_lat}[ak],$$

and the corresponding velocity components:

$$\text{velam}[ak], \text{vephi}[ak].$$

These can be extracted from the model-dependent parameter vector.

The length of the line segment $ds[ak]$ between points $ak - 1$ and ak is then derived by a call to the common function normal section length [and azimuth].

Convert the latitudes to radians:

$$\varphi_1 = \pi(\text{track_lat}[ak-1])/180.0 \quad \text{eq 5.3-20}$$

$$\varphi_2 = \pi(\text{track_lat}[ak])/180.0 \quad \text{eq 5.3-21}$$

$$\lambda_1 = \pi(\text{track_long}[ak-1])/180.0 \quad \text{eq 5.3-22}$$

$$\lambda_2 = \pi(\text{track_long}[ak])/180.0 \quad \text{eq 5.3-23}$$

Call the common procedure

$$\text{Normal section length}\{\varphi_1, \lambda_1, \varphi_2, \lambda_2; ds(ak), \beta\}$$

The required output is the line element $ds(ak)$.

Calculate the along-track distance by

$$s(0) = 0.0 \quad \text{eq 5.3-24}$$

$$s(ak) = s(ak-1) + ds(ak), \quad ak = 1, 2, \dots \quad \text{eq 5.3-25}$$

Re-define origin of y

For all ak

$$\text{track_y}(ak) = s(ak) - s(K)$$

Step 5.3.3.4.2 Generate Geolocation Grids

Starting with the tabular along-track latitudes and longitudes, the latitude and longitude and x-coordinate of every across-track tie point corresponding to the along track grid points are calculated, using spherical trigonometry. There are $N_1 + N_2 + 1$ tie points spaced by dL km. The following calculations are repeated for each value of $i = ak - K$, where $0 < i < \text{number of tie points in an orbit}$.

Step 5.3.3.4.2.1 Fill in central column of the grid

$$\text{grid_lat}(i, N_1) = \text{track_lat}[ak] \quad \text{eq 5.3-26}$$

$$\text{grid_long}(i, N_2) = \text{track_long}[ak], \quad \text{eq 5.3-27}$$

converted to lie in the range -180 to 180 by the subtraction (or addition) of 360 if required.

$$\beta' = \pi + \text{atan2}\{\text{velam}[ak], \text{vephi}[ak]\}. \quad \text{eq 5.3-28}$$

(Note that the azimuth given by the atan2 function should be a negative angle at this point).



Step 5.3.3.4.2.2 Fill in grid points to the east of of ground track

The latitudes and longitudes of N_2 points to the right of the sub-satellite point are calculated, using an interval of dL km between the points. These points are in the across-track direction, and lie on the normal section locally orthogonal to the sub-satellite track. The azimuth of this normal section is $\beta' - \frac{1}{2}\pi$:

$$\delta = \beta' - \frac{1}{2}\pi \quad \text{eq 5.3-29}$$

$$\varphi_1 = \pi(\text{track_lat}[ak])/180.0 \quad \text{eq 5.3-30}$$

$$\lambda_1 = \pi(\text{track_long}[ak])/180.0$$

For $j = (N_1 + 1), (N_1 + N_2)$

$$L = j * dL \quad \text{eq 5.3-31}$$

Use the common procedure Normal section endpoint to determine the latitudes and longitudes of the tabular grid point corresponding to j .

Normal section endpoint $\{\varphi_1, \lambda_1, L, \delta; \varphi_j, \lambda_j\}$

$$\text{grid_lat}(i, j) = (180/\pi)\varphi_j \quad \text{eq 5.3-32}$$

and their longitudes by

$$\text{grid_long}(i, j) = (180/\pi)\lambda_j. \quad \text{eq 5.3-33}$$

Each longitude should be converted to lie in the range -180 to 180 by the subtraction or addition of 360 if required.

Step 5.3.3.4.2.3 Fill in grid points west of ground track

$$\tilde{\varepsilon} = \beta' + \frac{1}{2}\pi \quad \text{eq 5.3-34}$$

$$\varphi_1 = \pi(\text{track_lat}[ak])/180.0 \quad \text{eq 5.3-35}$$

$$\lambda_1 = \pi(\text{track_long}[ak])/180.0$$

For $j = 0, 1, \dots, N_1 - 1$

$$L = 25 (N_1 - j) \quad \text{eq 5.3-36}$$

Use the common procedure Normal section endpoint to determine the latitudes and longitudes of the tabular grid point corresponding to j .

Normal section endpoint $\{\varphi_1, \lambda_1, L, \delta; \varphi_j, \lambda_j\}$ eq 5.3-37

The latitudes of the remaining tabular grid points are given by

$$\text{grid_lat}(i, j) = (180/\pi)\varphi_j \quad \text{eq 5.3-38}$$

and their longitudes by

$$\text{grid_long}(i, j) = (180/\pi)\lambda_j. \quad \text{eq 5.3-39}$$

Each longitude should be converted to lie in the range -180 to 180 by the subtraction or addition of 360 if required.

Note that a consistent sign convention is being adopted for the azimuths, so that the sign of $\delta [j]$ will be positive for eastward displacements, negative for westward.



5.3.3.5 Satellite Navigation and Attitude

A call to the CFI function `xo_osv_compute` is used to determine the satellite position and velocity vector, relative to the geostationary frame of reference, at the time t of each instrument pixel. This assumes that the initialisation has already been performed in step 5.3.3.4.1.

The geolocation algorithm as currently defined assumes that the satellite attitude model corresponds to yaw steering mode, and the orbit initialisation should have included this.

`status = xo_osv_compute(input parameters; output parameters);`

Input Parameters

<code>(&orbit_id</code>	From the initialisation above
<code>&mode</code>	Propagation model – dummy for current version
<code>&time_ref</code>	As above
<code>&time</code>	Reference time

Output parameters

<code>pos_out</code>	Osculating position vector at pixel time (Earth fixed CS)
<code>vel_out</code>	Osculating velocity vector at pixel time (Earth fixed CS)
<code>acc_out</code>	Osculating acceleration vector at pixel time (Earth fixed CS)
<code>ierr</code>	Status error vector

The satellite position vectors so derived will be used as inputs to subsequent calls to the `xp_targetinter` function.

5.3.3.6 Geolocate Pixels

A pixel is represented by a single sample of a single detector element, and we shall refer to each set of near-simultaneous samples of the detector elements of a single channel as a pixel group.

For each pixel p in each scan s the direction of the line of sight is determined in the scan reference frame. The corresponding direction cosines are determined, and transformed to the satellite reference frame, and thence to the geostationary reference frame.

The following transformation matrices are independent of the scan position (section 4.4.4.7).

The matrix M_{ab} enables one to go from the scan reference frame to the instrument reference frame.

K' is the scan_inclination_nadir defined in Geometry ADF

$$M_{ab}(-\kappa') = \begin{pmatrix} \cos \kappa' & 0 & \sin \kappa' \\ 0 & 1 & 0 \\ -\sin \kappa' & 0 & \cos \kappa' \end{pmatrix} \quad \text{eq 5.3-40}$$

The matrices M_z , M_y and M_x are misalignment matrices enabling one to go from the instrument frame to the platform:

$$M_z(\zeta) = \begin{pmatrix} \cos \zeta & \sin \zeta & 0 \\ -\sin \zeta & \cos \zeta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{eq 5.3-41}$$



$$M_y(\eta) = \begin{pmatrix} \cos \eta & 0 & -\sin \eta \\ 0 & 1 & 0 \\ \sin \eta & 0 & \cos \eta \end{pmatrix} \quad \text{eq 5.3-42}$$

$$M_x(\xi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \xi & \sin \xi \\ 0 & -\sin \xi & \cos \xi \end{pmatrix} \quad \text{eq 5.3-43}$$

The matrix M_{cm} enables one to take into account the reflection of the scan mirror in the scan reference frame:

$$M_{cm}^v = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\kappa_v / 2) & -\sin(\kappa_v / 2) \\ 0 & \sin(\kappa_v / 2) & \cos(\kappa_v / 2) \end{pmatrix} \quad \text{eq 5.3-44}$$

The index v identifies the view ($v = n|a = \text{nadir} | \text{along-track}$).
 κ_v is the scan_cone_angle defined in Geometry ADF.

The matrix M_{ps} enables to go from the satellite reference frame defined in the SRD to the reference one used in the CFI (old ENVISAT convention):

$$M_{ps} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad \text{eq 5.3-45}$$

The matrix M_{ps} represents a rotation of 180 degrees about the downward-pointing z axis to transform from the spacecraft reference frame to a frame (the platform frame) nominally aligned with the yaw steering frame and so having its x axis pointing forwards.

The initial directions of the lines of sight to the detector elements k in the respective scan frames are defined by the direction cosines $\{\lambda_k, \mu_k, \nu_k\}$. Define the matrix whose columns are the initial line of sight vectors. This matrix has 3 rows and k_{max} columns:

$$M_{los} = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_{k_{max}} \\ \mu_1 & \mu_2 & \dots & \mu_{k_{max}} \\ \nu_1 & \nu_2 & \dots & \nu_{k_{max}} \end{pmatrix} \quad \text{eq 5.3-46}$$

Note that there are two sets of vectors $\{\lambda_k, \mu_k, \nu_k\}$, one set containing two elements ($k_{max} = 2$) for the thermal channels at 1 km resolution, and one set containing eight elements ($k_{max} = 8$) for the short wave and visible channels. We believe that the same sets of vectors will be valid for both views, but this is subject to confirmation. The dependence on channel and (TBC) view is not shown explicitly here so as not to complicate the notation.

The following steps are carried out for each pixel on each scan $S_i = 0, INT_S, 2*INT_S, \dots$ in the general case these points are

$$\rho_i^n = 0, INT_P, 2*INT_P, \dots, MAX_NADIR_PIXELS - 1 \quad \text{eq 5.3-47}$$

on the nadir scan, and

$$\rho_i^a = 0, INT_P, 2*INT_P, \dots, MAX_ALONG_TRACK_PIXELS - 1 \quad \text{eq 5.3-48}$$



for the oblique scan.

Step 5.3.3.6.1 Determine line of sight and its direction cosines in the instrument reference frame.

Let ϕ_p be the scan angle of the pixel. The scan mirror encoder position taken from the spacecraft telemetry is converted into ϕ_p using the encoder_conversion_nadir and encoder_conversion_oblique parameters in the geometry ADF.

Define the matrix M_{ac} which represents the scan angle corresponding to each pixel

$$M_{ac} = \begin{pmatrix} \cos \phi_p & \sin \phi_p & 0 \\ -\sin \phi_p & \cos \phi_p & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad \text{eq 5.3-49}$$

The direction cosines of the lines of sight to the ground pixels after reflection at the scan mirror are given by the columns of the matrix

$$M'_{los} = M_{ac}^{-1} M_{cm}^{-1} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} M_{cm} M_{ac} M_{los} \quad \text{eq 5.3-50}$$

The direction cosines of the lines of sight relative to the platform reference frame are then given by the columns of the matrix

$$\begin{pmatrix} \lambda_1^p & \lambda_2^p & \dots & \lambda_{k_{\max}}^p \\ \mu_1^p & \mu_2^p & \dots & \mu_{k_{\max}}^p \\ \nu_1^p & \nu_2^p & \dots & \nu_{k_{\max}}^p \end{pmatrix} = M_{ps} M_x(\xi) M_y(\eta) M_z(\zeta) M_{ab}(-\kappa') M'_{los} \quad \text{eq 5.3-51}$$

for the nadir view, or

$$\begin{pmatrix} \lambda_1^p & \lambda_2^p & \dots & \lambda_{k_{\max}}^p \\ \mu_1^p & \mu_2^p & \dots & \mu_{k_{\max}}^p \\ \nu_1^p & \nu_2^p & \dots & \nu_{k_{\max}}^p \end{pmatrix} = M_{ps} M_x(\xi) M_y(\eta) M_z(\zeta) M'_{los} \quad \text{eq 5.3-52}$$

for the oblique view.

Here the nominal misalignment of the instrument to the spacecraft is a fixed value obtained from the pre-launch measurements of the instrument relative to the satellite. The default angular misalignments, ξ , η and ζ could be set to zero in which case there is no rotation. It shall be possible for the user to switch between the pre-flight fixed offsets as provided in the geometric characterisation ADF or use the geometric calibration model as given in the following step.

Step 5.3.3.6.2 Computation of the matrix to transforming the instrument frame to account for the pointing corrections from the geometric calibration model (see the TN "Geometric calibration model for SLST" (S3-MO-TAF-01054/2010) for an overview of the calibration principles).

For each view, there are N_{Qm} models (given as input), indexed by the date of the year, each being a quaternion LUT depending on a parameter X^v ($v = a$ for along-track view and n for nadir view). X^v is a vector of several parameters:

- the On Orbit Position (OOP),
- the satellite position in the earth reference frame
- the sun position in the earth reference frame



For the first O-GPP issue, the model will only depend on OOP, but the process described here can easily be generalized to several parameters.

For each view:

1. the quaternion LUT corresponding to the current date is determined by linear interpolation between quaternion LUTs stored in the data base).
2. the parameters composing X^V must be computed, possibly using CFI functions.
3. Computation of the quaternion $Q^V(X^V)$ corresponding to X^V :
 - a. SLERP interpolation at location X^V in the LUT obtained at step 1.
 - b. Normalisation of the quaternion $Q^V(X^V)$ to unity.
4. Finally $Q^V(X^V)$ is converted into the equivalent rotation matrix $T_{calib}^V(X^V)$ (following standard formulae).

The corrected direction cosines of the lines of sight relative to the spacecraft reference frame are then given by the columns of the matrix

$$M_{los}^S = \begin{pmatrix} \lambda_1^S & \lambda_2^S & \dots & \lambda_{k_{max}}^S \\ \mu_1^S & \mu_2^S & \dots & \mu_{k_{max}}^S \\ \nu_1^S & \nu_2^S & \dots & \nu_{k_{max}}^S \end{pmatrix} = T_{calib}^V(X^V) M_{los}^V \quad \text{eq 5.3-53}$$

for both views

The NAVATT packets also contain attitude quaternions (see section 5.4.3.2.1) and these can be treated in the same way described above.

Step 5.3.3.6.3 Use the CFI target subroutine to determine the latitudes and longitudes of the pixels

For the tie pixel in question, the outgoing line of sight direction corresponding to detector element k relative to the platform frame is given by column k of the above matrix, $\{\lambda_k^p, \mu_k^p, \nu_k^p\}$. We can derive a corresponding azimuth and elevation by inverting the equations

$$\lambda_k^p = \cos(elevation) \cos(azimuth) \quad \text{eq 5.3-54}$$

$$\mu_k^p = \cos(elevation) \sin(azimuth) \quad \text{eq 5.3-55}$$

$$\nu_k^p = \sin(elevation) \quad \text{eq 5.3-56}$$

or therefore

$$azimuth = \text{atan2}(\mu_k^p, \lambda_k^p) \quad \text{eq 5.3-57}$$

$$elevation = \text{atan2}(\nu_k^p, \sqrt{(\lambda_k^p)^2 + (\mu_k^p)^2}) \quad \text{eq 5.3-58}$$

If $azimuth < 0.0$ then

$$azimuth = azimuth + 360.0. \quad \text{eq 5.3-59}$$



In the above atan2 represents the arc tangent function of two arguments defined in the conventional way, atan2(y, x) being the angle whose tangent is (y/x) and whose quadrant is defined by the signs of x and y. Note that both azimuth and elevation are functions of detector index *k*.

For each tie pixel *p* and detector element index *k*, the azimuth and elevation just determined, together with the satellite position at the nominal time of the scan *s* as determined from the CFI orbit propagation subroutine and any attitude offset data, are used as parameters of a call to the CFI target subroutine to determine the latitude and longitude of the pixel.

This seems to need various stages of initialisation.

a) Create an empty attitude ID

```
status = xp_attitude_init(&attitude_id, ierr)
```

Input parameters: None.

Output parameters:

&attitude_id	New attitude structure
ierr	Status error vector

b) Populate it – this call uses the satellite position to define the attitude frame

```
status =xp_attitude_compute(parameters)
```

Input parameters:

&model_id	Model ID
&time_id	Structure that contains the time correlations
&sat_nom_trans_id	Structure that contains the Sat. Nom. Trans
&sat_trans_id	Structure that contains the Sat. Trans.
&instr_trans_id	Structure that contains the Instr. Trans.
&attitude_id	Structure that contains the Attitude (input/output)
&time_ref	Time reference ID
&time	Time in Processing Format
pos	Satellite position vector (Earth Fixed CS)
vel	Satellite velocity vector (Earth Fixed CS)
acc	Satellite acceleration vector (Earth Fixed CS)
&target_frame	Attitude FrameID

Output parameters:

&attitude_id	Modified input attitude structure
ierr	Status error vector

Call to xp_target_inter[section].

```
status = xp_target_inter(parameters)
```

Input Parameters:

&sat_id	Satellite ID
&attitude_id	Structure that contains the Attitude. (input/output)
&atmos_id	Structure that contains the atmosphere initialisation.



&dem_id	Structure that contains the DEM initialisation.
&deriv	Derivative ID - Allowed values: (0) XP_NO_DER (1) XP_DER_1 ST (2) XP_DER_2ND
&inter_flag	Flag for first or second intersection point selection Allowed values: (1) XP_INTER_1 ST (2) XP_INTER_2ND
&los_az	Azimuth of the LOS (Attitude Frame)
&los_el	Elevation of the LOS (Attitude Frame)
&geod_alt	Geodetic altitude over the Earth m
&los_az_rate	Azimuth-rate of the LOS (Attitude Frame)
&los_el_rate	Elevation-rate of the LOS (Attitude Frame)
&iray	Ray tracing model switch - Accepted values: (0) XP_NO_REF (1) XP_STD_REF (2) XP_USER_REF
&freq	Frequency of the signal Hz

Output parameters:

&num_user_target	Number of user defined targets calculated
&num_los_target	Number of LOS targets calculated
&target_id	Structure that contains the Target results
ierr	Status error vector

status = xp_dem_init(parameters)

Input parameters:

&mode	Digital Elevation Model initialization mode
&model	DEM initialization model (dummy in current implementation)
dem_file	File used for DEM initialization (See DEM Configuration file in [DAT_SUM])

Output parameters:

&dem_id	Structure that contains the DEM initialisation.
ierr	Status error vector

I think the logic is then that we call xp_target_inter with H = geod_ht = 0

We then call xp_target_extra_main if we want the ellipsoid co-ordinates, and call it again with a request for the DEM target.

status = xp_target_extra_main (parameters)

Input parameters:

&target_id	Structure that contains the Target results
&choice	Flag to select the extra results to be computed. Even though derivatives could be requested by user, they will not be calculated if the target was computed without derivatives (a warning is raised in this case)
&target_type	Flag to select the type of target; XP_USER_TARGET_TYPE XP_LOS_TARGET_TYPE XP_DEM_TARGET_TYPE



&target_number	Target number
----------------	---------------

Output parameters:

main_results	See [AD-4] table 263
main_results_rate	See [AD-4] table 263
main_results_rate_rate	See [AD-4] table 263
ierr	Status error vector

The required outputs for each pixel are taken from the output array main-results:

- Target geocentric latitude (Earth Fixed CS)
- Target geodetic latitude (Earth Fixed CS)
- Target geodetic altitude (Earth Fixed CS)
- Target to satellite pointing: Azimuth (Topocentric)
- Target to satellite pointing: Elevation (Topocentric)

We then need to call `xp_target_extra_target_to_sun` to determine the solar angles at the pixel.

```
status = xp_target_extra_target_to_sun(parameters)
```

Input parameters:

&target_id	Structure that contains the Target results
&choice	Flag to select the extra results to be computed. Even though derivatives could be requested by user, they will not be calculated if the target was computed without derivatives (a warning is raised in this case) XP_NO_DER XP_DER_1ST XP_DER_2ND
&target_type,	Flag to select the type of target; XP_USER_TARGET_TYPE XP_LOS_TARGET_TYPE XP_DEM_TARGET_TYPE
&target_number	Target number
&iray	Ray tracing model switch
&freq	Frequency of signal

Output parameters:

sun_results	Table 274
sun_results_rate	Table 274
sun_results_rate_rate	Table 274
ierr	Error vector

The required outputs for each pixel are taken from the array sun_results:



- Target to Sun (centre) azimuth (Topocentric CS)
- Target to Sun (centre) elevation (Topocentric CS)

5.3.3.7 Calculate Pixel x-y co-ordinates

The following steps (described in section 4.4.4.10) are carried out for every pixel. The ortho-geolocated latitudes and longitudes have been calculated for every pixel on the nadir scan as

$$p_t^n = 0, 1, 2, 3, \dots, MAX_NADIR_PIXELS - 1 \quad \text{eq 5.3-60}$$

and for the oblique scan

$$p_t^a = 0, 1, 2, 3, \dots, MAX_ALONG_TRACK_PIXELS - 1 \quad \text{eq 5.3-61}$$

in the representative case that $INT_P = 1$.

Step 5.3.3.7.1 Co-ordinates of pixel

The ortho-geolocated latitude and longitude of the pixel are

$$\varphi_p = (\pi/180.0) \cdot nadir_latitude(s_t, p_t^n) \quad \text{eq 5.3-62}$$

$$\lambda_p = (\pi/180.0) \cdot nadir_longitude(s_t, p_t^n) \quad \text{eq 5.3-63}$$

for the nadir pixels, and

$$\varphi_p = (\pi/180.0) \cdot inclined_latitude(s_t, p_t^a) \quad \text{eq 5.3-64}$$

$$\lambda_p = (\pi/180.0) \cdot inclined_longitude(s_t, p_t^a) \quad \text{eq 5.3-65}$$

for the oblique pixels. The longitudes should be reduced to the range (-180.0, 180.0) if necessary.

Step 5.3.3.7.2 Identification of the nearest tabular points

First, choose an initial value of the index ig from which to start the iteration.

```
Set interval_not_found = TRUE
```

```
Set iteration_number = 1
```

```
If first entry of scan initialise  $ig$ :
```

```
 $ig = K + (source\_packet\_ut\_time(s_t) + relative\_scan\_<view> - T0) / \Delta T$ 
```

where $relative_scan$ is table 6-31 of SY-4

```
While (interval not found)
```

Where K , $T0$ and ΔT are defined in section 5.3.3.4.

Step 5.3.3.7.3 Calculate azimuth of Q1Q2 (β_{12})

Calculate



$$\varphi_1 = (\pi / 180.0) \cdot \text{track_lat}(i_g) \quad \text{eq 5.3-66}$$

$$\varphi_2 = (\pi / 180.0) \cdot \text{track_lat}(i_g + 1) \quad \text{eq 5.3-67}$$

$$\lambda_1 = (\pi / 180.0) \cdot \text{track_long}(i_g) \quad \text{eq 5.3-68}$$

$$\lambda_2 = (\pi / 180.0) \cdot \text{track_long}(i_g + 1) \quad \text{eq 5.3-69}$$

Use the normal section length and azimuth procedure to compute the azimuth angle β_{12} . Note the length is not required.

Normal section length $\{\varphi_1, \lambda_1, \varphi_2, \lambda_2; s_{12}, \beta_{12}\}$

Note that s_{12} is an output of the procedure that is not used by subsequent processing.

The angle β_{12} will be in the range $0 \leq \beta_{12} < 180^\circ$, and will represent the azimuth of the track at Q_1 , measured anticlockwise from north.

Step 5.3.3.7.4 Calculate azimuth of Q1P (β_{1p})

Use the normal section length and azimuth procedure to compute the azimuth angle β_{1p} . Notepoint P as follows:

Normal section length $\{\varphi_1, \lambda_1, \varphi_p, \lambda_p; s_{1p}, \beta_{1p}\}$

Note that s_{1p} is an output of the procedure that is not used by subsequent processing.

Step 5.3.3.7.5 Iterate

This step describes how to locate the tabular interval in which the azimuthal angle between the ground track to the pixel changes through 90°

```

If abs ( $\beta_{12} - \beta_{1p}$ ) <  $\pi/2$  then
    direction = +1
else
    direction = -1
endif
If iteration_number = 1 then prev_dir = direction
If prev_dir = direction then
    ig = ig + direction
else
    interval_not_found = FALSE
iteration_number = iteration_number + 1
prev_dir = direction
end while
If direction = -1 then ig = ig + direction

```

The difference



$$\beta_{12} - \beta_{1p}$$

(reduced into the range $-\pi$ to π if necessary) represents the required intersection angle PQ_1Q_2 .

In order that the tabular point specified be the earlier of the two, if in the last step of the iteration the value of i_g was increased by 1, it is now decremented by one.

Once the correct interval has been found, the sign of difference $\beta_{12} - \beta_{1p}$, reduced into the range $-\pi$ to π if necessary, at either end-point determines on which side of the ground track the pixel P lies, and therefore the sign of the x co-ordinate of P. If $\beta_{12} - \beta_{1p}$ is positive, then the point P is to the right (east) of the ground-track and its x co-ordinate will be positive. If $\beta_{12} - \beta_{1p}$ is negative, then the point P is to the left (west) of the ground-track and its x co-ordinate will be negative.

Step 5.3.3.7.6 Calculation of the sides of the triangle

Once the correct interval has been identified, the co-ordinates are calculated by spherical trigonometry. Suppose that the point on the sphere whose co-ordinates are indexed by the final value of i_g from the iteration is Q_1 and that corresponding to $i_g + 1$ is Q_2 . The latitude and longitude of each point P, Q_1 and Q_2 are known, and therefore the length of each side of the triangle PQ_1Q_2 can be determined.

Update the co-ordinates of the ground track points:

$$\varphi_1 = (\pi / 180.0) \cdot track_lat(i_g) \quad \text{eq 5.3-70}$$

$$\lambda_1 = (\pi / 180.0) \cdot track_long(i_g) \quad \text{eq 5.3-71}$$

$$\varphi_2 = (\pi / 180.0) \cdot track_lat(i_g + 1) \quad \text{eq 5.3-72}$$

$$\lambda_2 = (\pi / 180.0) \cdot track_long(i_g + 1) \quad \text{eq 5.3-73}$$

Each of the remaining sides of the triangle PQ_1Q_2 are calculated in turn using this procedure, with subscripts replaced appropriately from the 3 indices {1, 2, p}.

Calculate the side Q_2P , s_{2p} ,

$$\text{Normal section length } \{\varphi_2, \lambda_2, \varphi_p, \lambda_p; s_{2p}, \beta\}$$

Calculate the side PQ_1 , s_{p1} ,

$$\text{Normal section length } \{\varphi_p, \lambda_p, \varphi_1, \lambda_1; s_{p1}, \beta\}$$

Calculate the side Q_1Q_2 , s_{12} ,

$$\text{Normal section length } \{\varphi_1, \lambda_1, \varphi_2, \lambda_2; s_{12}, \beta\}$$

The triangle is now fully determined and any of its angles can be found.

Step 5.3.3.7.7 The mean radius of curvature.

The remaining calculations will make use of Legendre's theorem. Compute a suitable mean radius of curvature.

$$\varphi_m = (\varphi_1 + \varphi_2 + \varphi_p) / 3 \quad \text{eq 5.3-74}$$

$$R = EARTH_MAJOR_AXIS \cdot (1 - e^2)^{1/2} / (1 - e^2 \sin^2 \varphi_m) \quad \text{eq 5.3-75}$$



Calculate the area of the triangle, and the spherical excess E :

$$s = (s_{12} + s_{2p} + s_{p1})/2 \quad \text{eq 5.3-76}$$

$$A = \sqrt{s(s - s_{12})(s - s_{2p})(s - s_{p1})} \quad \text{eq 5.3-77}$$

$$E = A/R^2 \quad \text{eq 5.3-78}$$

radians.

Step 5.3.3.6.8 The angle at Q

The next step is to compute the angle at Q_2 . (The angle at Q_1 could be used instead; the choice is arbitrary.) The angle of the plane triangle having sides of the same length is computed, and then corrected by the spherical excess. Note that we do not use the procedure `solve_spherical_triangle` at this point because it is an ellipsoidal triangle and not a spherical triangle.

If ($s_{2p} = 0$) then

(trap the special case of a triangle of zero area, in which the angle is indeterminate)

$$Q_2 = 0.0 \quad \text{eq 5.3-79}$$

else

$$s = (s_{12} + s_{2p} + s_{p1})/2 \quad \text{eq 5.3-80}$$

$$Q_2 = 2.0 \cdot \arctan \left\{ \sqrt{\frac{(s - s_{12})(s - s_{2p})}{s(s - s_{p1})}} \right\} + E/3 \quad \text{eq 5.3-81}$$

(end if)

Step 5.3.3.7.9 The x and y co-ordinates

Given the side Q_2P and the angle Q_2 , the right-angled triangle PQ_2X is fully determined, and the arcs PX and Q_2X can be calculated. The arc PX is determined by an application of the sine rule.

$$\sin PX = \sin (s_{2p}/R) \sin Q_2. \quad \text{eq 5.3-82}$$

The side Q_2X is determined from the tangent formula

$$\tan Q_2X = \tan (s_{2p}/R) \cos Q_2. \quad \text{eq 5.3-83}$$

The arc lengths ξ , ζ are determined by multiplying the angular lengths PX and Q_2X respectively by the mean radius of curvature determined above.

$$\xi = R (PX) \quad \text{eq 5.3-84}$$

$$\zeta = R (Q_2X) \quad \text{eq 5.3-85}$$

The sign of the x co-ordinate of p is determined by inspecting the sign of the angle $\beta_{12} - \epsilon$ derived at the last value of i_g above, reduced if necessary into the range

$$-\pi < (\beta_{12} - \beta_{1p}) < \pi \quad \text{eq 5.3-86.}$$

Then the x and y co-ordinates of the pixel, expressed in km, are

$$x = \text{sgn}(\beta_{12} - \beta_{1p}) \cdot \xi \quad \text{eq 5.3-87}$$

$$y = \text{track_y}(i_g + 1) - \zeta \quad \text{eq 5.3-88}$$

Assign the x and y co-ordinates to the relevant arrays:



$$\langle view \rangle_x(s_t, p_t^v) = x$$

eq 5.3-89

$$\langle view \rangle_y(s_t, p_t^v) = y$$

where $\langle view \rangle = nadir | along_track$, and where the superscript v represents n or a , as appropriate.



5.4 LEVEL 1A PROCESSING

Description of part 1 of the processing: the computation of the radiometric, spectral and geometric correction and calibration coefficients.

5.4.1 Algorithm Input

5.4.1.1 Level 0 Instrument Digital Numbers

5.4.1.2 Ancillary Radiometric Data

The following auxiliary input data sets are required by these processing stages:

1. Level 1B Processor Configuration File: defines processor configuration data, internal error codes and other items.
2. SLSTR Instrument Data File: defines source packet data structures, validation parameters and engineering unit conversions for the SLSTR instrument.
3. SLSTR Calibration Data File: defines temperature to radiance look-up tables and non-linearity correction tables for the SWIR channels where required.
4. Visible Calibration Data File: the Visible Calibration Coefficients ADS from a previous orbit.

Fuller details of the contents of these files are given in reference [RD-1] Annex A.

5.4.2 Processing Objective

5.4.2.1 Source Packet Processing (Stages 1 - 6)

The purpose of these steps is to unpack and validate the source packet data and convert it to engineering units for the higher processing levels that generate the products and geolocate the data. The main functions of these steps can be summarised as follows:

1. Source Packet Quality Checks: Performs basic quality checks on each raw packet, ensuring that only those that pass the checks contribute to the product.
2. Unpack Ancillary Data: Unpacks all required ancillary and housekeeping data including the temperatures of the on-board black bodies and instrument health and status information.
3. Validate Unpacked Ancillary Data: validates the unpacked ancillary and housekeeping data.
4. Convert Unpacked Ancillary Data: Converts to engineering units those items of ancillary data, including the temperatures of the on-board black bodies and other instrument temperatures and data items, for which a conversion to engineering units is defined.
5. Validate Converted Ancillary Data: Validates those converted ancillary data items that are essential to the calibration.
6. Science Data Processing: Unpacks and validates the science data containing the earth view and black body pixel counts for all available channels from each packet.

5.4.2.2 Infra-Red Channel calibration (Stage 7)

7. Infra-red Channel Calibration: This step calculates the calibration offset and slope that describe the linear relationship between pixel count and radiance for the thermal IR channels. The parameters are determined from the black body pixels counts and the black body temperatures, and the process makes use of look-up tables for the conversion of temperature to radiance.



5.4.2.3 Visible Channel calibration (Stage 8)

8. Visible Channel Calibration: This step unpacks the viscal data once per orbit (if present), when it is detected that the viscal unit is in sunlight, and calculates calibration parameters for all visible channels. These calibration parameters may not be used to calibrate the science data in the visible channels for the current orbit (which may make use of VISCAL data from a previous orbit), but are written to the Visible Calibration Coefficients ADS in the GBTR product.



5.4.3 Mathematical Description

5.4.3.1 Source Packet Processing

Section 5.1 has described the unpacking of the Level 0 product and/or the Instrument Simulator output into a data structure that essentially consists of a two-dimensional array of structures `ISP_data[i][j]`, indexed by scan number `i` and ISP type `j`. These data structures form the starting point for Level 1 processing.

We have identified the basic processing stages in Section 5.2.1 and Figure 5.2.

5.4.3.1.1 Source Packet Quality Checks (Stage 1)

Basic validation checks are performed on the source packets in order to identify major problems such as data corruption which would require the affected source packet to be excluded from subsequent processing. In the case of SLSTR it is also necessary to identify and flag missing source packets, since for example the calibration targets are only sampled once in every two scans, while the visible channels are not sampled at night.

In the present formulation, these tests have been performed during the stage of data input and unpacking described in Section 5.1.2, and their results can be found in the array of flag structures `quality[i][j]` described there.

A single overall validity flag for the ISP of type `j` corresponding to scan `i` can be defined as the union (logical OR) of the individual flags:

$$ISP_invalid[i][j] = \bigcup_k quality[i][j][k]$$

5.4.3.1.2 Housekeeping Packet processing (Stages 2-5)

The housekeeping source packet (PCAT = F) contains a variety of parameters relating to the SLSTR subsystems, including quantities such as black body temperatures required for STSTR Level 1 processing. The contents of the packet are defined in RD-5.

Let us define the following notation for the ISP structures

`ISP[i][j].record[k]`

`HK_packet[i].record[k]`

where both record structures are byte arrays indexed by `k`. Here `i` indexes the scan and `J` the ISP type. Obviously `HK_packet[i].record[k] = ISP[i][‘HK’].record[k]`, where ‘HK’ is the ISP type associated with the Housekeeping packet.

Individual parameters occupy fields of variable length in the record (and do not start on, say, even byte boundaries). For Level 1b processing, each relevant field must be unpacked and the contents verified and where relevant converted to engineering units. Each parameter should be identified by a unique parameter identifier to act as a cross-reference between the ancillary data tables used to control the processing. AATSR processing used a 5-character alphanumeric code for this purpose, and this method has been adopted in the tables here, but logically any unambiguous convention could be adopted.

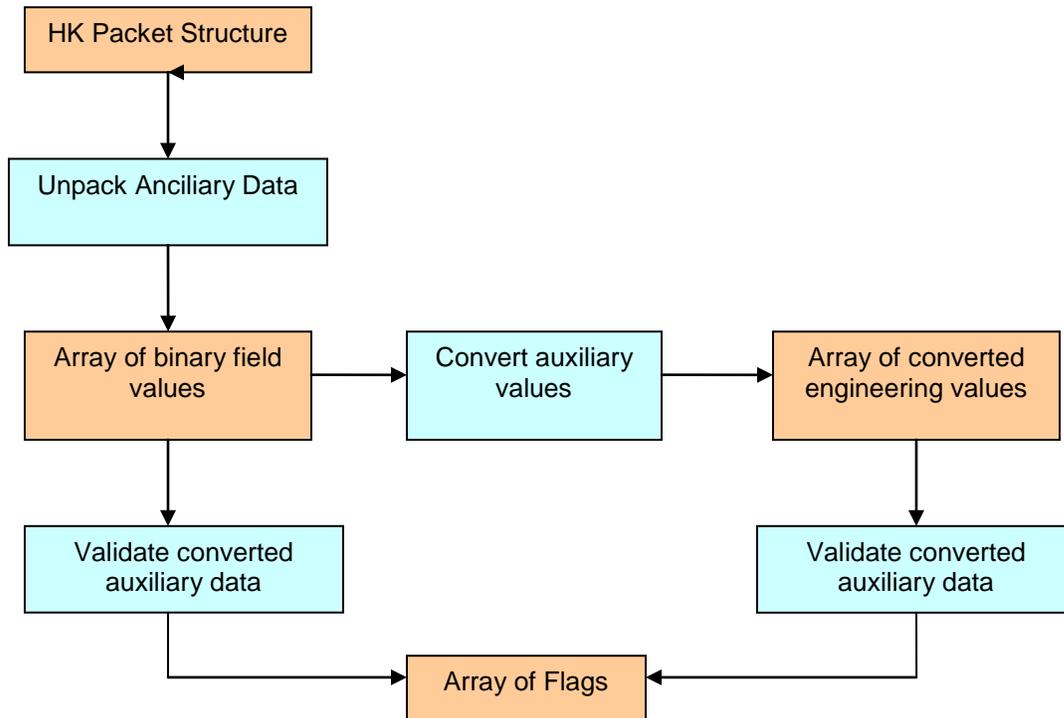


Figure 5-6

5.4.3.1.3 Ancillary Data Unpacking (Stage 2)

In order to accommodate the variable field lengths in which the parameters are stored, a generic table-driven algorithm (similar to those used for AATSR) is proposed to unpack these items and where necessary to convert them to engineering units.

For maximum generality we allow for the possibility that an item of data may occupy only part of a field, although there may be few if any such cases defined in RD-5.

The SLSTR telemetry specification will define the length and starting byte within the ISP of each field, and if applicable, the starting bit number and length of each item within the field. This information can be stored in the Ancillary Data ADF which defines the field in the source packet in which each variable can be found, and if applicable the mask required to extract the variable from the field, and the shift required to correctly justify it. This table is then read in at the start of processing and used to unpack each ancillary data item from the raw packet, storing the resultant integers in an array.

The table will contain one entry for each parameter from the House-keeping source packet that is required by the Level 1b processing algorithms. The total number of entries is defined by the ancillary parameter *anc_tot*. Each entry will contain the following;

- Ancillary item identifier
- Start byte of field in ISP
- Length of field in bytes
- Mask to extract ancillary item from field
- Shift to justify extracted ancillary data item.



For each scan i , and for each ancillary data item l , the corresponding ancillary datum is then unpacked as follows.

```
ancdata(l) = HK_packet[i].record[offset:offset+item_length-1] AND  
extraction_mask(i)  
  
if bit_shift[l] ≠ 0 then shift ancdata(i) left by bit_shift[l] bits
```

For this version of the ATBD the relevant fields in the Ancillary ISP are

- Byte 12 – Gain S1-S3
- Byte 35-44 – Adjustable Bias for DU4-DU7
- Byte 102-119 – Detector Temperatures
- Byte 138-201 – BB PRTs and reference resistors
- Byte 256-299 – TAE0 PRTs
- Bytes 318-445 – TAE0 TH_1 to TH_32

5.4.3.1.4 Validation of Unpacked Ancillary Data (Stage 3)

The validation of the unpacked ancillary data consists of limit checks on the unconverted data. Limit checking is table driven; the upper and lower limits of all specified unpacked ancillary items are read in from a data file along, with the identifier with which they are associated. The result of each validation is loaded into the integer array `ancillary_data_validation_result` which is defined so that the same index is used to access corresponding values from `ancdata` and from `ancillary_data_validation_result`.

For each ancillary datum for which limit checks are defined, the corresponding data set will contain upper and lower limit values `validation_lower[i]`, `validation_upper[i]` defined in the Ancillary data ADF. These will be provided in a data record tagged with the Ancillary item identifier.

Then for each $k \in \{k: \text{limit parameters for ancillary data item } k \text{ are defined}\}$

```
If ancdata[k] < validation_lower[k] or ancdata[k] > validation_upper[k]  
Then ancillary_data_validation_result[k] += 1 (Validation Error)
```

This error code indicates that the item is out of range. Note that it is assumed that the array `ancillary_data_validation_result[k]` is initialised to zero.

5.4.3.1.5 Conversion of Unpacked Ancillary Data (Stage 4)

Not all ancillary items can be converted to engineering units. For example, some items may be status flags for which no conversion is sensible. If a conversion is defined for an item, the conversion function will be specific to that item, and may not be known until ground characterisation of the instrument. For this reason, the conversion of the unpacked ancillary data is specified in the Ancillary Data ADF.

The ADF describes, for each variable that can be converted, which of a number of generic conversion functions to use, and the list of parameters to use for the conversion. A series of generic functions are defined to cater for each mathematical operation likely to be required in the conversion of SLSTR engineering data.

For each ancillary datum, the ADF entry will contain the following fields:

- the function identifier (`function_ID`): this specifies the function to be used to perform the conversion for this item.



- The number of parameters in the conversion function (num_param)
- the list of parameters (parameter) associated with the function.

For example, if the conversion function associated with a given parameter is

$$y = (z + a) / b$$

where z is the unconverted ancillary data value, then a and b are the parameters associated with the conversion of this datum. Note that the same function may be used for the conversion of different ancillary data values but with different parameters.

In some cases it may be necessary to specify other, unpacked telemetry items in the list of function parameters, as well as constants. These parameters may be distinguished from constants by the adoption of some suitable convention. For example, such items might occur first in the parameter list, and be distinguished from constants by an identifying prefix of "A".

The values of the converted data items are loaded into a floating point array converted_ancillary_data. All ancillary items that are not eligible for conversion shall be cast to floating point and copied to this array, so that all ancillary items are accessible in a single array, rather than two.

The ancillary ADF of conversion parameters will contain n_{conv} records for each parameter, which we can index by k ($0 \leq k < n_{anc}$). Denote the contents of the record as follows:

```
item_ID[k]
function_ID[k]
a[k], b[k], ...= conversion parameters
```

For each ancillary data item identify the conversion function function_ID[k] and corresponding parameters a[k], b[k], ...

Then provided that the unpacked item is valid, that is if ancillary_data_validation_result[k] ≠ 1 then the item is converted using the specified conversion function

```
conv_anc_data[k] = fn(unpacked_ancillary_data[k], a[k], b[k], ...)
```

5.4.3.1.6 The generic conversion algorithms

For each conversion algorithm, f_0 to f_{14} , Z is the integer value of the unpacked ancillary data item which is being converted, and a , b , c , d , e , f , g and h ,...etc. are the values of parameters which are supplied specifically for each ancillary data item which uses that function. In more complex functions, "result" is the name of a local variable in which the result of the function is returned.

If a conversion is not necessary for a parameter, it has a function ID= f_0 .

```
f0:          float(Z)

f1:  if (Z=0) then 4
     else if (Z=1) then 8
     else if (Z=2) then 16
     else if (Z=4) then 32
     else if (Z=8) then 64
     else if (Z=9) then 128
```



f2: $a + bZ$

f3: $a + bZ + cZ^2$

f4: $a + bZ^2 + cZ^2 + dZ^3$

f5: $a^Z * b$

f6: aZ/b

f7: $(Z+a)/b$

f8: $a+(Z/b)$

f9: $a/(Z+b)$

f10: if $(Z \geq a)$ then $b+cZ$
else $d+eZ$

f11: $((aZ+b)^2)/c$

f12: if $(Z < a)$ then
 $cZ+d$
else if $(a \leq Z \leq b)$ then
 $eZ+f$
else
 $gZ+h$

f13: Black body temperature function

According to the BBEU design description, the BBEU stores 32 words each containing the value corresponding to a sampled PRT or resistor output. Each word contains a sequence number as well as relating to the temperature range.

Bits 1-12 – converted analogue value for input channel

Bits 13-17 word number in binary

Bits 21-24 temperature range setting

Words 2-9 correspond to the PRTs and Resistors in the Hot BBC

Words 10-17 correspond to the PRTs and Resistors in the Cold BBC

Based on the information provided in the BBEU calibration and characterisation plan (S3-PL-ABS-SL-0016) and design description (S3-RP-ABS-SL-0010) the conversion logic from digital to analogue values is



Convert digital value to voltage

$$V = b_0 + b_1D + b_2D^2 \text{ (TBC SG)}$$

Convert voltage to temperature using polynomial of the form

$$T = a_0 + a_1V + a_1V + a_2V^2 \text{ (TBC SG)}$$

Where a separate set of coefficients a_n are used for each temperature range setting given in bits 21-24.

f14: TAE0 temperature conversion functions

According to the BBEU design description, the BBEU stores 48 words each containing the value corresponding to a sampled PRT or thermistor output. Each word contains a sequence number as well as relating to the temperature range.

Bits 1-12 – converted analogue value for input channel

Bits 13-17 word number in binary

Bits 21-24 temperature range setting

Words 2-9 correspond to the PRTs

Words 17-18 correspond to the thermocouple

The conversion logic from digital to analogue values is TBD pending information from SG

It is expected that the conversion from digital number to temperature will use the following logic

Convert digital value to voltage

Convert voltage to temperature using polynomial of the function

$$T = a_0 + a_1D + a_1D + a_2D^2 + a_3D^3 + a_4D^4 + a_5D^5 \text{ (TBC SG)}$$

Where a separate set of coefficients a_n are used for each temperature range setting.

f20: Look-Up Table

Here 2 x Nparameter LUT is provided in the ADF giving converted values as a function of Z. The conversion from Z to anc_conv_data is performed by a polynomial interpolation function.

5.4.3.1.7 Validation of Converted Ancillary Data (Stage 5)

Further quality assurance is provided by tests on the converted values of selected parameters from the HK packet. There are two types of test; the first are general limit checks on appropriate values, the second are additional specific tests on those values that are particularly important for calibration, e.g. BB temps. The design of the instrument is not sufficiently advanced to enable these tests to be specified in detail. However, we here set out tests based on the equivalent AATSR tests to indicate the level of complexity involved. These specialised tests are performed principally on the data of each housekeeping packet which has passed the packet quality check.

5.4.3.1.7.1 General limit checks



This test is applied to all converted items for which maximum and minimum surveillance limits are specified in the ancillary data ADF. If no limit is specified, then no check is made. Limits are specified as the surveillance limits, which contains one record for each parameter. Denote the contents of the record as follows:

- ancillary identifier item_ID[k]
- surveillance lower bound - surveillance_lower[k]
- surveillance upper bound - surveillance_upper[k]

For each ancillary data item k the value is compared with the surveillance limits.

If((conv_anc_data[k] < surveillance_lower[k]) OR (conv_anc_data[j] > surveillance_upper[k]))

then

logical OR ancillary_data_validation_result array[k] with error_code

Where error_code = 4 if conv_anc_data[k] is one of the black body temperature values (BB Temp out of limit), otherwise error_code = 2.(TM Surveillance error)

5.4.3.1.7.2 Specific tests on critical parameters

As noted above, the design of the instrument is not sufficiently advanced to enable these tests to be specified in detail. Here we set out tests based on the equivalent AATSR tests.

Check of the paraboloid mirror temperatures and the black body PRT counts

This test is performed on the unconverted counts of the black body temperature sensors (BB PRTs and reference resistors) and of other critical temperature transducers (TAE0 PRT and TAE0 Th) to be specified.

For each count to be tested, if the value is either zero or max_count then

logical OR the ancillary_data_validation_results array with the error_code at the corresponding index. k, for that sensor.

Where error code = 8 (Indicates BB PRT raw value out of range)

The error code will ensure that the value is not included in the calibration calculations. The value max_count is the maximum count possible for the field (equal to $2^n - 1$ for an n-bit field).

Test of the Paraboloid mirror temperatures transducers

The test uses the converted values of the paraboloidal mirror temperatures transducers (TAE0 PRTs). If none of the telemetry values have been flagged in the corresponding element of the ancillary_data_validation_results array, the arithmetic mean of the valid temperature values is calculated:

$$mean = \frac{1}{n} \sum_{k=1}^n T_k$$

where n is the number of temperature transducers contributing to the sum and T_k is the temperature of the k'th transducer.

The value of each sensor in turn is compared with the mean. If the absolute difference exceeds prt_max_difference (defined in the additional parameters in the Ancillary data ADF), the datum is flagged.



If $abs(T_k - mean) > prt_max_difference$ then flag this datum by a logical OR with the error_code into the corresponding field of the ancillary_data_validation_results array

Where error_code = 16 (Indicates rogue PRT sensor)

Black body over range check.

This test depends on the existence of an over-range flag associated with each black body temperature transducer.

For each black body temperature value, if the error code in the corresponding element of the ancillary_data_validation_result array indicates a valid conversion and if the corresponding over range bit for that sensor is set, then flag that black body sensor datum as not to be used for the calibration. If this occurs then

Then logical OR the error_code into the corresponding element of the ancillary_data_validation_result array.

Where error_code = 32 (Indicates BB Overrange Error)

Check each black body sensor temperature

This test is carried out provided that the appropriate telemetry status flag is set to indicate that the heater is selected for one of the black bodies. If this is not the case, the black body data cannot be used to determine the calibration of the thermal channels; If this occurs then

logical OR the error_code into the corresponding element of the ancillary_data_validation_result array.

Where error_code = 64 (Indicates Rogue BB temperature sensor)

The heated black body is determined by comparing the temperatures of BB1 and BB2. The black body with the highest temperature is then the heated one.

If the heated black body is BB1 then

BB1_PRT_MAX_DIFF = (MAX_HBB_PRT_DIFF)

BB2_PRT_MAX_DIFF = (MAX_CBB_PRT_DIFF)

else

BB1_PRT_MAX_DIFF = (MAX_CBB_PRT_DIFF)

BB2_PRT_MAX_DIFF = (MAX_HBB_PRT_DIFF)

The following test is then applied to each black body separately. A weighted mean temperature is calculated for each black body, and the separate temperature values (BB PRT readings) are compared with this. Any that deviate by too large an amount are flagged as defective.

Step 1. Calculate the weighted mean temperature of both hot and cold black bodies.

Note. The purpose of the weighted mean algorithm is to incorporate redundancy, so that the mean value can be derived, even if one (or more) sensors fail.

Denote the number of temperature sensors associated with the black body under consideration by n , and their outputs by T_k , $0 < k \leq n$. We define a valid value as one for which the error code in the corresponding element of the anc_data_validation_result array indicates a valid conversion. We divide the set of valid temperature values into two subsets $K1$, $K2$ on the basis of their geometric layout. The thermometers should be arranged so that there is a group of 3 near the centre of the black body



baseplate, two towards the edge and one on the baffle. To get the correct weighting we need to group the three on the centre together and the two on the edge. We do not use the baffle sensor in the average. The precise location of each sensor is TBD. Then the weighted mean temperature BB1 is calculated as follows.

$$mean_1 = \frac{1}{n_1} \sum_{k \in K_1} T_k$$

$$mean_2 = \frac{1}{n_2} \sum_{k \in K_2} T_k$$

where n_1 and n_2 are the numbers of valid temperatures included in the respective sums.

If we now define $n_3 = 2$ if both sums are valid (i.e. $n_1 > 0$ and $n_2 > 0$), or $n_3 = 1$ if either $n_1 = 0$ or $n_2 = 0$, then the weighted mean temperature is given by

$$mean = (mean_1 + mean_2) / n_3$$

Note that if $n_1 = n_2 = 0$ no valid temperatures have been recorded and the calibration cannot proceed.

Step 2. Test each sensor value on the BB1.

For each sensor k on BB1 calculate the absolute difference

$$abs_diff = abs(T_k - mean)$$

between the temperature of the sensor and the weighted mean BB temperature derived above. If abs_diff is greater than $BB1_PRT_MAX_DIFF$ then

logical OR the `error_code` into the corresponding element of the `anciliary_data_validation_result` array.

Where `error_code = 64` (Indicates Rogue BB temperature sensor)

to flag the sensor value as not to be used for the calibration.

Step 3. Test each sensor value on the BB2.

The test is then repeated for BB2.

Test on reference resistor.

The following test is to check that the BB reference resistor values measured by the BBEU are at the expected values. If the corresponding `anciliary_data_validation_result` indicates a valid conversion then if the corresponding element of the array `conv_anc_data` is outside the range $REFERENCE_RESISTOR_VAL \pm REFERENCE_RESISTOR_VARIANCE$, then

logical OR the `error_code` into the corresponding element of the `anciliary_data_validation_result` array.

Where `error_code = 128` (Indicates Reference Resistor Error)

Note. The error code added is here not enough on its own to qualify the temperature for exclusion from the calculation of the calibration parameters, but it may be added to an existing error code, the combined error may take the error value over the threshold for inclusion.

Test the Detector Temperature Values.

Step 1 Test the converted values of IR channel detector temperatures (S7-S9 detector temperatures). Derive the maximum difference max_diff between the temperatures converted from the IR channel detector temperatures (that is, the range spanned by the temperatures).



For each of the converted temperatures taken individually, if the corresponding ancillary_data_validation_result error code indicates a valid conversion and if max_diff is greater than IR_DET_TEMP_DIFF then

logical OR the error_code into the corresponding element of the ancillary_data_validation_result array.

Where error_code = 256 (Indicates Detector Temperature Error)

(Note. The error code added is not enough on its own to qualify the temperature for exclusion from the calculation of the calibration parameters, but since the error for this condition may be added to an existing error code, it is possible that the combined error may take the error value over the threshold for inclusion.)

Step 2 Test the visible channel detector temperatures

Repeat the test for the visible channel detector temperatures (S1-S6 detector temperatures), but replacing the test by max_diff > VIS_DET_TEMP_DIFF.

Scanner and Flip Mirror Error Checks

The quality checks for the scanners and flip mirrors will utilise the encoder positions for each data sample provided in the scan ISPs. The SCAN mirrors and Flip Mirror encoder positions are frozen on every PIX10SYNC occurrence. The SUE will produce a data packet every PIX10SYNC period containing the positions acquired (at the beginning of) two PIX10SYNC periods before. See figure 5-7 below.

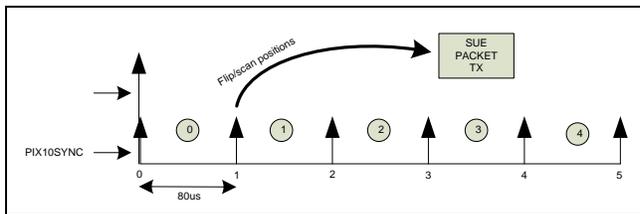


Figure 5-7

The error checks assume that the scanners and flip mirrors are controlled in position by the Scan Unit Electronics (SUE) according to a nominal control law [RD-2].

The scanners will follow two ideal position profiles (one for each mirror) synthesised by the SUE (using the PIX10SYNC and SCANSYNC) and starting from two programmable start positions NA_ZERO_POS and OB_ZERO_POS. I.e. the actual positions as given by the encoder values at each SCANSYNC occurrence correspond to the values respectively stored in NA_ZERO_POS and OB_ZERO_POS registers.

The flip mirror performs a complete switch cycle (NA -> OB -> NA -> OB -> NA) in a CYCLESYNC period (600ms). The timing of the flip transitions is programmable by TC by means of four parameters containing the four flip transition times expressed in multiples of PIX10SYNC during a CYCLESYNC period (i.e. 4 numbers in the range 0 to 7339)

Using the encoder positions we can test for



- Noise errors (bearing, encoder noise)
- Deviation from nominal position

Results of the scan and flip mirror error checks are written to the pointing_flags confidence word of the global ADS.

Bit	Text code	Meaning if set	Comment
0	FlipMirrorAbsoluteError	flip mirror absolute error exceeds threshold	
1	FlipMirrorIntegratedError	flip mirror integrated error exceeds threshold	
2	FlipMirrorRMSError	flip mirror RMS error exceeds threshold	
3	ScanMirrorAbsoluteError	scan mirror absolute error exceeds threshold	
4	ScanMirrorIntegratedError	scan mirror integrated error exceeds threshold	
5	ScanMirrorRMSError	scan mirror RMS error exceeds threshold	
6	ScanTimeError	Scan time is inconsistent with scan count sequence	
7	Platform_Mode	platform mode	0 if nominal, else 1

Scan Mirror Tests

The basic principle of this test is to compare the actual encoder value against the nominal value used for the control law (or based on a long term average). The input for this test shall be a table of expected scan mirror position versus acquisition number (assuming 80us per Pix10Sync). The outputs of the test are flags for each view (nadir, oblique) defined as follows

- Scan Mirror Absolute Error (per pixel), if set to 1 if the absolute difference between the actual and expected value for the pixel exceeds a threshold value of TBD arcseconds.
- Scan Mirror Integrated Error (per view), is set to 1 if the sum of the Absolute pixel values exceeds TBD arcseconds.
- Scan Mirror RMS Error (per view), is set to 1 if the RMS of the absolute pixel errors exceeds TBD arcseconds.

The flags are written to pointing flags in the GlobalFlags ADS (ref)

The test is performed over the nadir and oblique views for each SCANSYNC period as follows

```
InstantaneousError = double[TargetLength[iview]];
TotalPositionError = 0;
SumsqPositionError = 0;
NumSamples = 0;

for(i=0, i<TargetLength[iview], i++){
    ExpectedScanValue(i) = target first acquisition + i
```



```

InstantaneousError[i] = ScanEncoderValue[i,iview] -
ExpectedScanValue[i,iview];

ScanMirrorAbsoluteError[i] = (abs(instantaneousError[i]) >
InstantaneousThreshold);

TotalPositionError += InstantaneousError[i];
SumSqPositionError += PositionError[i,j]*PositionError[i,j];
NumSamples++;
    }
}

ScanMirrorIntegratedError = (Abs(TotalPositionError) > TotalThreshold);

```

The result of this test should of course be zero. A non-zero value > 1pixel would indicate that the scan control has not been able to maintain the speed of the mechanism and could indicate a control loop error or mechanism roughness. This is equivalent to the ATSR jitter test except that it is applied only to the view and not the complete scan.

```

AvPositionError = TotalPositionError/NumSamples;
RmsPositionError = Sqrt(SumSqPositionError/NumSamples -
AvPositionError*AvPositionError);

ScanMirrorRMSError = (RmsPositionError >RMSThreshold);

```

Flip Mirror Tests

The basic principle of this test is similar to the scan mirror test and compares the actual encoder value against the nominal value at each position. Since the flip transition occurs outside the data collection window, this will not be recorded. However, the test should show that the flip mirror is at the expected position and the stability. The outputs of the test are flags for each view (nadir, oblique) defined as follows

- Flip Mirror Absolute Error (per pixel), is set to 1 if the absolute position exceeds TBD arcseconds.
- Flip Mirror Integrated Error (per view), is set to 1 if the average position exceeds TBD arcseconds.
- Flip Mirror RMS Error (per view), is set to 1 if the RMS error exceeds TBD arcseconds.

The flags are written to pointing flags in the GlobalFlags ADS (ref)

The test is performed over the nadir and oblique views for each SCANSYNC period and for each scanner as follows

```

InstantaneousError = double[TargetLength[iview]];
TotalPositionError = 0;
SumSqPositionError = 0;
NumSamples = 0;

for(i=0, i<TargetLength[iview], i++){
    InstantaneousError[i] = FlipEncoderValue[i,iview] -
    Flip_nominal_position[iview];
}

```



```
FlipMirrorAbsoluteError[i] = (abs(instantaneousError[i]) >
InstantaneousThreshold);

TotalPositionError += InstantaneousError[i];
SumsqPositionError += PositionError[i,j]*PositionError[i,j];
NumSamples++;
}
}
```

```
FlipMirrorIntegratedError = (Abs(TotalPositionError) > TotalThreshold);
```

The result of this test should of course be zero. A significant non-zero value would indicate that the flip mirror is slowly drifting in position during the scan cycle.

```
AvPositionError = TotalPositionError/NumSamples;
RmsPositionError = Sqrt(SumsqPositionError/NumSamples -
AvPositionError*AvPositionError);
```

```
FlipMirrorRMSError = (RmsPositionError >RMSThreshold);
```

Scan Time Inconsistency Check

This checks that the time in the scan time ISP is consistent with the scan count. Here we compare the actual time in the ISP against the expected time in the ISP based on the previous scan time.

```
Expected_time = time(scan-1) + 300ms.
ScanTimeErr = (ISP time stamp ≠ Expected_time ± 80ms);
```

Platform mode Check

The platform mode refers to the pointing guidance mode as contained in the NAVATT packets (see section 3.7 of RD-14) which is given once every second. Therefore, the pointing mode needs to be interpolated for each scan and then mapped onto the image grids during re-gridding.



5.4.3.2 Science Data Processing (Stage 6)

This stage of processing extracts the detector signals and any associated variables from the instrument Source Packets.

We assume that the instrument source packets have been read from the Level 0 product and assembled into the data arrays described in Section 5.1.2. (This merely requires that the Level 0 processing steps described in that section are reversed. Then for each ISP type described in Table 5.1 (excluding the Housekeeping ISP), the relevant data is read into memory.

We adopt the following notation for the extracted channel counts

Following Section 5.1.2.1. we designate the content of the source packet of type j corresponding to scan number i by $content[i][j]$. The detector count corresponding to given pixel will be denoted by

$$C[ch, target; S, p, t, k]$$

where ch is an index to the channel, $target$ represents the target type (Table 5-3), and the remaining indexing is that introduced in Section 4.2.3. The necessary indexing is complex because a given instrument pixel is associated with a given channel, target and detector as well as a particular scan and acquisition.

Then the procedure is as follows.

```
for each scan  $i \in \{i; 0 \leq i < n\_scan\}$ 
  for each ISP type  $j \in \{j; j \text{ corresponds to a valid channel and target ID}\}$ 
    scansync[i][j] = content[i][j].scansync
    validity[i][j] = content[i][j].validity
    target_id[i][j] = content[i][j].target_id
    target_first_acq[i][j] = content[i][j].target_first_acq
    target_length[i][j] = content[i][j].target_length
```

Extract the pixel counts. The details depend on the channel corresponding to the ISP type.

For the visible channels (S1, S2, S3)

$$C[ch, target_id; S, p, t, k] = content[i][j].pixel_value[p][t][k]; ((k = 0, 3), t = 0, 1), p = 0, target_length - 1)$$

For the SWIR (TDI) channels (S4, S5, S6)

$$C[ch, target_id; S, p, t, k] = content[i][j].pixel_value[p][t][k]; ((k = 0, 7), t = 0, 1), p = 0, target_length - 1)$$

For the thermal infra-red and fire channels (S7, S8, S9, F1, F2)

$$C[ch, target_id; S, p, k] = content[i][j].pixel_value[p][k]; ((k = 0, 1), p = 0, target_length - 1)$$



5.4.3.2.1 NAVATT Packet processing

The NAVATT data set is included in the Level 0 product. It comprises an XML header and two binary files; an annotations file, and a file of NAVATT Source Packets. In normal operation NAVATT packets are generated at a 1 Hz rate and comprise satellite position and attitude information sampled at intervals of 1 second, together with time correlation information.

NAVATT packet processing comprises unpacking each ISP into an element of an indexed data structure containing the orbit and attitude data for each sample.

The data fields to be unpacked are shown in the following table.

Field #	Field content
1	GPS time
2	OBT
3	Time correlation validity flag
4	Time of NAVATT data samples
5	Satellite X co-ordinate
6	Satellite Y Co-ordinate
7	Satellite Z co-ordinate
8	Satellite X velocity
9	Satellite Y velocity
10	Satellite Z velocity
11	Attitude Quaternion #1
12	Attitude Quaternion #2
13	Attitude Quaternion #3
14	Attitude Quaternion # 4
15	Attitude Quaternion Difference (Est. – Target) #1
16	Attitude Quaternion Difference (Est. – Target) #2
17	Attitude Quaternion Difference (Est. – Target) #3
18	Attitude Quaternion Difference (Est. – Target) #4
19	Pointing Mode
20	Manoeuvre Flag
21	GNSS Status Flag
22	Orbit Revolution Number
23	Orbit position angle (OOP) – unit $360/2^{32}$ degrees.

These data may be used as input to the time correlation process and to the CFI calls for orbit and target positioning. Note that because of the possibility missing or invalid data, (as evidenced by the presence of validity flags) additional processing is required at this stage to interpolate any missing elements. This processing should be common to the SLSTR and OLCI processing.



5.4.3.3 Infra-red channel calibration

The calculation of the calibration parameters involves determining of the slope and offset for each channel, detector element, pixel parity and each of the nadir and oblique views separately, from the unpacked and validated data in the processed telemetry source packets. It averages the necessary values over the number of packets in a 'calibration interval'; this is a short interval, of the order of 10 source packets (TBD). The calibration parameters so derived are then used to calibrate all the packets in the calibration interval.

In each scan cycle the two black bodies are each viewed once each through each view optics chain (nadir and oblique). It follows that the minimum possible length of a calibration period will be one scan cycle (CYCLESYNC interval), and in practice a calibration interval must be an integer number of scan cycles. The length of the calibration interval will be specified in the Level 1b Processor Configuration Ancillary Data Product. If the scans are numbered by S , and if the origin of S is defined so that the first scan of a scan cycle is denoted by an even value of S , then $S/2$ is an index to the scan cycles. We denote the first and last scans of a calibration interval by S_1 , S_2 . Note that while in general the calibration interval is of a fixed length, it is possible that the gain or offset values for the channel may change for some S in $S_1 < S < S_2$; this condition should be checked, and if it arises, the value of S_2 should be reset to the scan number of the last scan to which the old values applied.

It is stated in section 5.4.3.2 that the counts from the TIR ISPs use the following notation:

$$C(\text{ch}, \text{target_id}; S, p, k)$$

Where ch denotes the channel (S6-S9, F1-2), the target_id denotes the target viewed as defined in table 5-3, S is the scan count, p is the pixel ($0 < p < \text{target_length} - 1$) and k is the detector element ($k=0$ or 1 for the TIR).

When calibrating, the parity of the pixels need to be distinguished. In the case of the TIR channels, the odd and even pixels occur every other pixel acquisition and therefore the counts from the instrument can be represented as follows:

$$C(\text{ch}, \text{target_id}; S, \text{px}, t, k) = C(\text{ch}, \text{target_id}; S, p, k)$$

For the IR channels we use t to denote the pixel parity. Where if p is even, $t=0$ to denote an even pixel, and if p is odd, $t=1$ to denote an odd pixel. In this case, px is used to represent the pixel acquisitions of each parity and the size of px will be half the target length for that particular target.

The following five steps are applied to all packets in a calibration period for all valid pixels.

Step 1. Derive mean BB temperatures.

The weighted mean thermometer values for each black body (BB) are calculated for both hot and cold black bodies, from all valid values from one packet. This is repeated for all packets in the calibration period. The component means are summed, and a single mean is derived for each BB over the calibration period.

The weighted mean temperatures, T_{BB1} and T_{BB2} are calculated using the equations in Section 5.4.3.1.7.

The mean background (fore-optics) temperature, T_{inst} is also derived from valid data over the same period.



Step 2. Convert the mean BB temperatures to radiance.

Convert each of the weighted mean PRT temperatures for each BB to a radiance value for each IR channel. This is achieved by means of a look up table converting temperature to radiance, with one look up table for each IR channel. The radiance for each black body $L_{BB1}(ch)$ and $L_{BB2}(ch)$ is then a function of channel ch .

The mean instrument temperature is also converted to radiance for each of the TIR channels. $L_{inst}(ch)$.

The look-up tables are derived by integrating the Planck function over the channel filter profile for each channel. All look up tables are adjusted for any detector non-linearity determined during ground characterisation of the instrument to ensure a linear relationship between the input and output. Conversion is likely to be considerably more efficient if the data are supplied at sufficiently close enough intervals to allow linear interpolation between points, rather than having fewer data points and relying on polynomial interpolation.

Step 3. Correct the BB radiance values.

Each black body radiance value is now corrected to derive the true mean black body radiance L_{true} , using the mean background radiance L_{inst} derived in step 2, together with emissivity constants $\epsilon_{BB1}(ch)$, and $\epsilon_{BB2}(ch)$.

$$L_{true, BB1}(ch) = \epsilon_{BB1}(ch) L_{BB1}(ch) + (1 - \epsilon_{BB1}(ch)) L_{inst}(ch)$$

$$L_{true, BB2}(ch) = \epsilon_{BB2}(ch) L_{BB2}(ch) + (1 - \epsilon_{BB2}(ch)) L_{inst}(ch)$$

Step 4. Derive the mean BB pixel count for each BB over the calibration period.

The mean count values are calculated for odd and even pixels, for both BB1 and BB2, and for all channels and detector elements, from all valid BB pixels in the calibration period. First, the mean (odd and even) pixel count for each channel is derived from the valid BB pixel data in each packet. This is repeated for all packets in the calibration period, and the component means are summed. From this sum of means, a single mean is derived to give a mean BB pixel count over the calibration period, for odd and even pixels, for each BB, for each detector element and for each IR channel.

The pixel sums are calculated independently for each channel, based on the Instrument Source Packets referring to the black body targets. For each channel there are four such ISPs, corresponding to target IDs as follows [RD-6].

Black Body	Nadir view scan	Oblique view scan
BB1	B0	B1
BB2	C0	C1

Table: Target ID (hexadecimal) corresponding to each black body and view combination, from Reference [RD-6].

Denote the black body pixel counts from the relevant ISP by $C_{bb}(ch, target_id, S, px, t, k)$, where $target_id$ represents the target ID of the black bodies (as in the table above), s the scan cycle, pa is the pixel parity, px is the pixel ($px0 < px < px1$, $px0 = target_first_acquisition$, $px1 = target_first_acquisition + target_length$ for each target id) is an index to the pixel samples for a given detector, and k denotes the detector. (For the thermal and fire channels there are of course two detector elements, and $k = 0, 1$)



Then the mean channel count for the target denoted by ID and detector k is

$$\bar{C}_{bb}(ch, target_id, t, k) = \frac{1}{N} \sum_{s=s1}^{s2} valid(s) \sum_{px=px0}^{px1} C_{bb}(ch, target_id, S, px, t, k)$$

where N is the number of samples that contribute to the sum.

The mean count should also be calculated for the cold black body for the solar channels, to provide the calibration offset for these channels.

It will be necessary to determine from the telemetry (presumably the HK ISP) which of BB1 and BB2 is the cold and which the hot black body. How this is done is TBD pending the availability of a detailed telemetry specification.

Step 5. Calculate the gains and offsets from BB counts and BB radiance.

Derive the calibration_slope and the offset for the calibration period for nadir and oblique views, using:

$$slope_{IR}\{ch, view, t, k\} = \frac{L_true_{BB1}(ch) - L_true_{BB2}(ch)}{\bar{C}_{bb}\{ch, ID_1, t, k\} - \bar{C}_{bb}\{ch, ID_2, t, k\}}$$

$$offset_{IR}\{ch, view, t, k\} = \frac{L_true_{BB1}(ch)\bar{C}_{bb}\{ch, ID_2, t, k\} - L_true_{BB2}(ch)\bar{C}_{bb}\{ch, ID_1, t, k\}}{\bar{C}_{bb}\{ch, ID_2, t, k\} - \bar{C}_{bb}\{ch, ID_1, t, k\}}$$

where $ID_1 = B0$ or $B1$ for nadir or oblique views respectively
 $ID_2 = C0$ or $C1$ for nadir or oblique views respectively.

Step-6 Calculate the radiometric noise from the blackbody signals

First we compute the signal channel noise $\sigma C_{bb}(ch, target_ID, pa, k)$, which is taken as the standard deviation of the detector counts C .

Calculate the mean detector counts measured over the calibration period:

$$C\{ch, target_ID, px, t, k\} = \frac{1}{N} \sum_{s=s1}^{s2} C\{ch, target_ID, s, px, t, k\}$$

Where N is the total number of valid scans that contribute to the sum.

Then the signal noise is calculated as follows:

$$\sigma C_{bb}\{ch, target_ID, t, k\} = \left(\frac{1}{n-1} \sum_{px=px0}^{px1} (C\{ch, target_ID, px, t, k\} - \bar{C}_{bb}\{ch, target_ID, t, k\})^2 \right)^{1/2}$$

where



target_ID is B0, B1, C0 or C1

C_{bb} is the mean blackbody pixel count calculated in step 4.

n is the number of pixels summed

We then compute the differential of radiance wrt temperature, $\partial L/\partial T$ at each blackbody temperature T_{bb} using the temperature to radiance look-up table (as in step 2) so that

$$\left. \frac{\partial L(ch, target_ID)}{\partial T(target_ID)} \right|_{T_{bbx}} = \frac{L(ch, T_{bbx} + 0.1) - L(ch, T_{bbx})}{0.1}$$

When x=1, the target_ID=B0 or B1 for BB1

x=2, the target ID=C0 or C1 for BB2.

For the infrared channels the instrument noise is expressed as the noise equivalent brightness temperature difference (NEDT) for each black body,

$$dT_{BB}\{ch, target_ID, t, k\} = slope_{IR}\{ch, view, t, k\} \times \left(\sigma C_{bb}\{ch, target_ID, t, k\} \left(\left. \frac{\partial L(ch, target_ID)}{\partial T(target_ID)} \right|_{T_{bb}} \right)^{-1} \right)$$

Where for view = nadir, target_ID=B0 or C0 for BB1 and BB2 respectively

for view=oblique, target_ID=B1 or C1 for BB1 and BB2 respectively.

As with step 4, NEDT values are calculated for odd and even pixels, for both hot and cold black bodies, and for all IR channels and detector elements, from all valid BB pixels in the calibration period.



5.4.3.4 Solar channel calibration

The solar channel gain parameter is derived from the pixel counts over the visible calibration target (the VISCAL unit) during the period when it is illuminated by the sun, and the illumination level is constant. The illumination period is determined and the average and standard deviation of all valid pixels from the VISCAL unit for all scans during the calibration period so determined are calculated. A similar mean and standard deviation are calculated for all valid pixels from a selected black body over the same scans. The latter defines the dark signal. This is done for each channel to be calibrated.

It is stated in section 5.4.3.2 that the counts from the VIS and SWIR ISPs use the following notation:

$$C\{ch, target_id, S, p, t, k\}$$

Where *ch* denotes the channel (S6-S9), the *target_id* denotes the target viewed as defined in table 5-3, *S* is the scan count, *p* is the pixel ($0 < p < target_length - 1$), *k* is the detector element ($k = 0 - 3$ for VIS, $k = 0 - 7$ for SWIR) and *t* defines the acquisition cycle (which is equivalent to parity).

When calibrating, the parity of the pixels need to be distinguished. In the case of the VIS and SWIR channels, as there are two acquisition cycles in PIX10SYNC period, the parity alternates so there will be an odd and even pixel for each cycle.

so

$$C\{ch, target_id, S, px, t, k\} = C\{ch, target_id, S, p, t, k\}$$

Where *t* indexes the acquisition cycle of the pixel. $t = 0$ for 1st PIX05SYNC acquisition and $t = 1$ for 2nd PIX05SYNC acquisition. Here $px = p$ is the pixel number as defined by the PIX10SYNC period.

This section describes the process for extracting the calibration data for the SLSTR visible channels using the on-board calibration system.

Step 1. Select window around VISCAL illumination period.

Calculate the time when the VISCAL is under full solar illumination. This time is denoted by *calibration_time* and expressed in days, as given in the formula below.

$$calibration_time = ascending_node_time + (orbit_period * (270 - solar_declination_angle) / 360 + time_offset) / 86400.$$

where

ascending_node_time is the ascending node crossing time

orbit_period is the Sentinel-3 orbit period ~100 minutes

time_offset is the period between the time when the VISCAL is fully illuminated and the time that Sentinel-3 crosses the terminator, ~ 4.8 TBD minutes. (The time of full illumination occurs after the terminator (day-night boundary) crossing.)

solar_declination_angle is the solar declination angle

solar_declination_angle =

$$(0.006918 - 0.399912 * \cos(day_angle)) +$$



$$\begin{aligned}
 &0.070257 * \sin(\text{day_angle}) - \\
 &0.006758 * \cos(2 * \text{day_angle}) + \\
 &0.000907 * \sin(2 * \text{day_angle}) - \\
 &0.002697 * \cos(3 * \text{day_angle}) + \\
 &0.00148 * \sin(3 * \text{day_angle})) * 180 / \pi
 \end{aligned}$$

and

$$\text{day_angle} = 2 * \pi * (\text{day} - 1) / 365$$

is the day angle for the nth day of the year (lqbal 1983).

To ensure that all the data for the VISCAL is captured when illuminated by the Sun, a window of \pm TBD minutes around calibration_time should be used. Thus

$$\begin{aligned}
 t_start &= \text{calibration_time} - \text{window_half_width_in_min} / 1440 \\
 t_end &= \text{calibration_time} + \text{window_half_width_in_min} / 1440
 \end{aligned}$$

where t_start and t_end are expressed in days, and where

$$\begin{aligned}
 \text{window_half_width_in_min} &= \text{TBD} \\
 1440 &= \text{number of minutes in a day}
 \end{aligned}$$

Step 2. Process the raw data for the time window between t_start and t_end.

Calculation of solar calibration parameters can only be done if the window centred on the calibration time computed in Step 1 falls within the product limits. Although this should be true for consolidated data, if NRT data is being processed it is possible that the start or end of the Level 0 data will fall within the VISCAL illumination period. In this case it may not be possible to derive a valid visible channel calibration. In particular,

if t_start is earlier than the time of the first scan it may not be possible to generate a valid VISCAL packet, and VISCAL processing should be abandoned.

Otherwise, and provided there are at least 34 TBD scans present in the period between the start of the VISCAL window (t_start) and the end of the Level 0 data process the raw data to extract the pixel counts and instrument telemetry as follows.

[Implementation note:

The algorithm as defined here does not cater for the possibility of a null or invalid scan falling within the time window. In the present prototype implementation, such scans are omitted entirely from the VISCAL processing. The valid scans that fall within the time window determined in Step 1 are copied to a scratch file, which is then re-read and processed as in the following steps.

Experience suggests that such invalid scans are rare, and the omission of one or two sporadic scans (say for a CRC failure) will not significantly bias the algorithm, but the presence of a data gap might lead to an erroneous calibration that would not be recognized as such.

If the algorithm following is applied to a block of scans including invalid scans, the question of how to interpret the invalid monitor count will arise. If the invalid monitor count were set to zero, the effect of a single missing scan would be small in most cases, but the behaviour of the algorithm in the presence of the data gap is not currently specified.]



Step 3. Find period within the time window (t_{start} and t_{end}) when the smoothed VISCAL monitor is above the specified threshold

In AATSR, monitoring of the VISCAL unit was performed using a photodiode whereas for SLSTR the pixel counts from one of the channels will be used to monitor the brightness of VISCAL. By default, the monitoring channel will be the $0.87\mu\text{m}$ channel, although there should be flexibility for this to be altered and this is allowed for by including the parameter `detection_channel` in the SLSTR specific auxiliary data file. The detector element to use for monitoring is `detection_element` in the SLSTR specific auxiliary data file.

Define `monitor_count[view; s]` as average counts from 0.87 (or otherwise) channel averaged over all pixels as a function of scan cycle s .

Here, we find the period of time when the monitor channel counts are above a certain threshold value (`monitor_threshold`) to ensure the VISCAL is in full solar illumination. The `monitor_threshold` is equivalent to the `detection_threshold` in the auxiliary data file.

Several tests are performed during this process resulting in confidence flags `found_start[view]` and `found_end[view]` to indicate valid VISCAL start and end data has been found.

The indices of the first and last scans respectively in the time window identified in step1 are denoted by `first_s[view]` and `last_s[view]`

If `last_s[view]` is the index of the last scan in the Level 0 product being processed, so that end of data falls within the time window, it may not be possible to generate valid VISCAL parameters. The tests below involving `last_s[view]` allow for this case.

Initialise flags:

```
found_start[view] = FALSE
found_end[view]   = FALSE
```

If there are fewer than `detection_scan_threshold` scans present in the period between the start of the VISCAL window and the end of the Level 0 data (this situation might arise for NRT data) an initial value of the smoothed monitor count, `mon1`, cannot be derived. In this case a valid VISCAL product cannot be generated, and solar calibration parameter processing should be abandoned.

If $last_s - first_s < detection_scan_threshold$ then abandon.

Otherwise, we smooth the monitor count data measured between `first_s` and `last_s` in order to find the correct centroid of the illumination.

Let the smoothing interval be `sm_int`

```
sm_int = 1
```

The index of the first scan to be processed is not `first_s` but `s0`, where

```
s0 = (first_s + sm_int)
```



Calculate a smoothed monitor count `mon1` for the first sample where the centre of the interval is given more weighting:

```
mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0
```

where `s` is greater than or equal to `s0`

If `mon1 > monitor_threshold`, it may not be possible to derive a valid centroid, and processing should be abandoned.

Otherwise do the following while statements so that we find the value of `s` when `mon1 > monitor_threshold`:

```
while mon1 < monitor_threshold and s0 < s < (last_s - sm_int)
s = s + 1
mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0
end while
```

When the while loop ends, it defines the first value of `s` of the scan cycles for which the smoothed monitor count exceeds the threshold. We also find the `source_packet_ut_time`, the UTC time of packet `s`.

```
found_start = TRUE
s1 = s (start of monitor window)
ut1 = source_packet_ut_time(s) (monitor window start time)
```

Note that `s` will correspond to the index of the first sample of a block of 8. The next loop can skip in increments of `sm_int`.

```
while mon1 ≥ monitor_threshold and s ≤ (last_s - 2*sm_int)
s = s + sm_int
mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0
end while
```

If `mon1 < monitor_threshold`, the while loop has terminated because of the condition on `last_s`, and it may not be possible to define a valid centroid, because there is no more data in the monitor time window. This case may arise if the end of the Level 0 product falls within the time window. In this case a valid VISCAL slope cannot be generated, and processing should be abandoned.

Otherwise, the last scan cycle for which the smoothed monitor count exceeds the threshold is:

```
found_end = TRUE
s2 = (s - sm_int)
ut2 = source_packet_ut_time(s) (monitor window end time)
```

Note. This is to be sure that the data used corresponds to the time when the VISCAL is being illuminated by the Sun and not in darkness.

Step 4. Find period of full solar illumination



Now calculate the centroid of the monitor count values between s_1 and s_2 .

```
sum0 = 0.0 d0
sum1 = 0.0 d0
for s = s1, s2, sm_int do
    sum0 = sum0 + monitor_count[s]
    sum1 = sum1 + s * monitor_count[s]
end for
mean_s = integer part of [sum1/sum0]
```

Define the start of the calibration period for the signal channels as being *calibration_window_diff1* scans before the first moment of the of the monitor signal.

```
n1 = mean_s - calibration_window_diff1
```

Define the end of the calibration period for the signal channels as being *calibration_window_diff2* scans after the first moment of the monitor signal.

```
n2 = mean_s + calibration_window_diff2
```

The values n_1 and n_2 determined above define the limits of the calibration period for the following steps. Thus the length of the calibration window is

```
calibration_window_diff1 + calibration_window_diff2 + 1
samples.
```

Calculate an estimate of the mid-point of the monitor window

```
mid_UT_time = (ut1 + ut2)/2.
```

Step 5. Average Pixel Counts over Calibration Period.

The next stage is to calculate the average and standard deviations of the VISCAL and black body counts over the calibration period defined in the previous step. The cold black body is used in this case to determine the visible calibration coefficient.

Note that at this point we expect that $n_1 \geq \text{first_s}$ and $n_2 \leq \text{last_s}$ otherwise this step cannot be carried out. In this situation (which can only arise in the event of a significant gap in the data) a valid VISCAL parameter cannot be generated, and Steps 5 to 7 should be abandoned. Otherwise:

Derive the average pixel count and standard deviation for all channels and for all scans which see full illumination as follows:

$$\bar{C}_{viscal}(ch, view, t, k) = \frac{1}{M} \sum_{px=m1}^{m2} \frac{1}{N} \sum_{s=n1}^{n2} C(ch, target_ID, S, p, t, k)$$



Where target_ID=D0h when view=nadir and target_ID=D1h when view=oblique and pa indexes the acquisition cycle of the 05PIXSYNC channels.

Identify which of the black bodies is the cold black body and get the average counts as a function of channel (where only S1-S6 are calculated), parity pa, detector element k and nadir and oblique view.

If $T_{BB1} < T_{BB2}$ then

$T_{BBC} = T_{BB1}$ and
Target_IDc = B0_h for view = nadir
Target_IDc = B1_h for view = oblique

else

$T_{BBC} = T_{BB2}$ and
Target_IDc = C0_h for view = nadir
Target_IDc = C1_h for view = oblique

$$\bar{C}_{bbc}(ch, view, t, k) = \frac{1}{M} \sum_{px=m1}^{m2} \frac{1}{N} \sum_{s=n1}^{n2} C(ch, target_IDc, S, p, t, k)$$

where

m_1 and m_2 are the start and stop pixels for the view;

M is total the number of pixels used in the view

n_1 and n_2 are the start and stop scans of the calibration window,

and

N is the number of valid scans in the calibration window between n_1 and n_2 inclusive that contribute to the calibration.

Step 6. Calculate Calibration slope

For each channel ch , the calibration slope for the solar channels is then determined using

$$slope_{sol}(ch, view, t, k) = \frac{Reflectance_factor(ch, k) \times Gain(ch)}{\bar{C}_{viscal}(ch, view, t, k) - \bar{C}_{bbc}(ch, view, t, k)}$$

where:

Gain[ch] is the instrument gain setting for channel ch – for SLSTR there are three coarse gain settings of $x0.5$, $x1$ and $x2$ with the default being $x1$.

Reflectance_factor[ch , k] is the reflectance of the VISCAL diffuser at the channel wavelength; and

C_{viscal} is the average pixel count from the VISCAL system,

C_{bbc} is the average pixel count from the cold black body previously determined.



This assumes a linear relationship between detector counts and source reflectance. If a non-linear correction has been determined for the channel during pre-launch characterisation, a correction is applied to the detector voltages before the entire calibration process is started and so the assumption of linear calibration will always be true.

Step 7. Reflectance to radiance conversion parameter

The seasonally adjusted solar irradiance, weighted by the channel filter profile, is also calculated for each solar channel to enable the conversion from reflectance to radiance (Section 3.5.3.8) is also calculated.

$$\begin{aligned} \text{solar_irradiance}[ch, \text{view}, k] &= \text{mean_solar_irradiance}[ch, \text{view}, k]^* \\ & (1.000110 + 0.034221 * \cos(\text{day_angle}) + 0.001280 * \sin(\text{day_angle}) \\ & + 0.000719 * \cos(2 * \text{day_angle}) + 0.000077 * \sin(2 * \text{day_angle})) \end{aligned}$$

where

$$\text{day_angle} = 2\pi(\text{day_in_year} - 1) / 365$$

and where the values of *mean_solar_irradiance* are taken from the ancillary file of configuration data. We can also calculate the VISCAL radiance, *L_viscal*

$$L_viscal(ch, \text{view}, k) = \text{Reflectance factor}(ch, \text{view}, k) \times \text{solar_irradiance}(ch, \text{view}, k) / \pi$$

Step 8. Calculation of noise

Finally we compute the signal channel noise of the VISCAL signal σC_{viscal} , which is taken as the standard deviation of the detector counts for the VISCAL during the illumination period and is defined by

$$\sigma C_{viscal}(ch, \text{view}, t, k) = \left(\frac{1}{M} \sum_{px=m1}^{m2} \frac{1}{N} \sum_{s=n1}^{s=n2} (C_{viscal}(ch, \text{view}, s, p, t, k) - \bar{C}_{viscal}(ch, \text{view}, t, k))^2 \right)^{1/2}$$

Where

target_ID=D0h when view=nadir and target_ID=D1h when view=oblique

C_{VISCAL} is the count for channel ch (S1-S6), pixel *px* of parity *pa*, detector element *k* and view,

and \bar{C}_{VISCAL} is the mean VISCAL pixel count calculated in step 4.



$$\sigma C_{bbc}(ch, view, t, k) = \left(\frac{1}{M} \sum_{px=m1}^{m2} \frac{1}{N} \sum_{s=n1}^{s=n2} C(ch, target_IDc, S, p, t, k) - \left(\frac{1}{M} \sum_{px=m1}^{m2} \frac{1}{N} \sum_{s=n1}^{s=n2} C(ch, target_IDc, S, p, t, k) \right)^2 \right)^{1/2}$$

Where target_IDc relates to the target ID of the cold black body defined earlier, in the appropriate view.

$$dL_viscal(ch, view, t, k) = \frac{\sigma C_{viscal}(ch, view, t, k) * slope_{sol}(ch, view, t, k) * solar_irradiance(ch)}{\pi}$$

$$dL_bbc(ch, view, t, k) = \frac{\sigma C_{bbc}(ch, view, t, k) * slope_{sol}(ch, view, t, k) * solar_irradiance(ch)}{\pi}$$

The values are calculated for odd and even pixels for channels S1 to S6, from all valid VISCAL pixels in the illumination period.



5.5 LEVEL 1B PROCESSING

This section describes the processing at Level 1b, including the calibration of the instrument pixels, the re-sampling of the L1b data onto a suitable image grid, and the processing that relates to the regridded data.

The description follows the AATSR practice in which the specified image grid is related to the satellite ground track and instrument swath. Note also that the AATSR grid used equal time interval sampling in the along track direction; that is, although the grid was sampled at 1 km in the across-track direction, in the along-track direction the sampling interval was equal to the distance moved in one instrument scan, which varies around the orbit. If alternative sampling were proposed, then modifications to the scheme would be required.

Again following AATSR practice, we have included land/sea flagging and cloud clearing as processes following regridding.

5.5.1 Algorithm Input

5.5.1.1 Level 1a Data

5.5.1.2 Auxiliary Data

The following auxiliary input data set is required by the signal channel calibration processing stage:

- SLSTR Calibration Data File: defines radiance to temperature look-up tables and non-linearity correction tables for the SWIR channels where required.

The following auxiliary input data sets are required by this cloud clearing processing stage:

- Land-Sea Mask Data File.
- Cloud LUT Data File: contains look-up tables used by the cloud clearing tests.

Fuller details of the contents of these files are given in reference [RD-1] Annex A.

5.5.2 Processing Objective

5.5.2.1 Signal Calibration (Stage 17)

Uncalibrated scan pixels are calibrated using calibration coefficients derived earlier. Pixel calibration uses these calibration coefficients to convert the pixel data to brightness temperature, in the case of the infrared channels, or to radiance in the case of the visible and near-visible channels.

5.5.2.2 Time Domain Integration (Stage 18)

In order to provide resolutions of 500 m in the short-wave channels, the design of the current FPA for these channels incorporates a 2 by 4 array of detectors, the long edge being aligned in the along-track direction. As a consequence, the swath is effectively scanned twice in the across-track direction, once by each column of the detector array, and so it is appropriate to perform averaging on these duplicate samples to reduce noise. Note that at the swath edge the pixels of the resultant averaged image will be distorted, but the individual plines will be retained to support level 1c processing.

This step follows signal calibration so that the averaging is applied to calibrated reflectances. This is because the calibrations of the detectors in the two columns may differ.



5.5.2.3 Regrid Pixels (Stage 19)

Calibrated AATSR pixels are regridded into co-located oblique and nadir images, onto a 1 km grid using the pixel positions derived previously. The process used in (A)ATSR processing has migrated instrument pixels to the nearest grid point. This is an input-driven process, in the sense that the process loops over input pixels, and is a highly efficient process in computational terms, but it has been suggested that an output-driven process (looping over image pixels) would present advantages. If such an algorithm were adopted, the next process (cosmetic fill) would not be required.

5.5.2.4 Cosmetic Fill (Stage 20)

Cosmetic filling of nadir/oblique view images is performed, to fill missing image pixels.

5.5.2.5 Image Pixel Positions (Stage 21)

Linear interpolation is performed to determine the latitude and longitude coordinates of the grid pixels for use in the subsequent stages.

5.5.2.6 Determine Land-Sea Flag (Stage 22)

Given the image pixel latitude and longitude, the surface type for each image pixel is derived using the land/sea flagging algorithm.

5.5.2.7 Cloud clearing (Stage 23)

The cloud-clearing algorithms are used to identify image pixels as cloudy or cloud-free. Baseline AATSR processing uses nine independent tests based on the AATSR suite of channels, and has recently been expanded by the addition of a new visible channel test. The additional channels proposed for SLSTR will permit additional tests yet to be specified.

5.5.2.8 Meteo annotations (stage 24)

Algorithm Inputs

Level 1b data

- Pixel geo-referencing data: geographic location, viewing and illumination geometry.

Meteorological data are provided by models, as a global prediction of a future situation (hereafter called global forecast) as well as a global view of a past situation, consolidated with observation data such as in situ measurements and remote sensing data (global analysis). These forecasts and analysis are provided by meteo centres like the ECMWF at regular time sampling, typically every 6 hours, spatially sampled on regular geographic grids.

The best available quality level shall be selected at the time of processing, i.e. forecast for NRT and analysis for NTC.

Since the time sampling of meteo files do not match the Sentinel-3 acquisition, temporal interpolation is required and meteo files shall be available at the two closest time samples. Some of the fields require values before and after the nominal time in 6 hourly intervals for the diurnal thermocline modelling and so 5 values are required per analysis time i.e. -18, -12, -6, 0, +6hrs.

Algorithm outputs

The main outputs of this processing step are the annotations to Level 1b products corresponding to meteorological data estimated at the subset of the product pixels. Acquisition geometry, i.e. illumination and viewing angles, are also provided.

Global per product quality flags are generated to indicate the level of quality of the input files (forecast or analysis) and the availability of the appropriate meteo time samples.



5.5.3 Mathematical Description

5.5.3.1 Calibration of Pixel Data

For each channel and for each instrument pixel, the raw pixel counts is converted to brightness temperature (for the thermal and fire channels) or to reflectance (for the solar channels) by the application of the calibration relationship. In all cases a linear relationship is assumed between measured count and either radiance or reflectance as appropriate.

The pixel counts have already been unpacked into a structure described in section 5.4.3.3 as

$$C\{ch, target_id, S, px, t, k\}$$

As we only need the nadir and oblique Earth targets the counts can be represented by the following:

$$C_{scene}\{ch, view, S, px, t, k\}$$

where view = nadir if target_id is A0hex, and view=oblique if target_id is A1hex.

Every scan is calibrated, using the set of slope and offset values that were calculated closest in time to the scene scan. The slopes and offsets previously determined were for calibration period measured between scans S1 and S2. The slope and offset values are now represented as a function of scan count S where it is assumed the slopes and offset calculated between S1 and S2 are copied into the array for all S where $S1 \leq S \leq S2$.

$$\begin{aligned} &Slope\{ch,view, S,t,k\} \\ &Offset\{ch,view, S,t,k\} \end{aligned}$$

For channels S1-S6 the slope_{sol} parameter is used, for channels S7-S9 and F1-2, the slope_{IR} and offset_{IR} parameters are used.

The calibration is performed for all pixels px of each parity, where the pixels relate to the nadir or oblique targets. Note for the solar irradiance channels, the pixel parity is identical to the acquisition sequence.

For each view pixel and for each detector we have a corresponding exception byte that is copied to the MDS. The definition of the exception byte is given in SY-4 and is reproduced as follows.

Bit Number	Text Code	Description
0	ISP_absent	ISP absent
1	pixel_absent	Pixel absent
2	not_decompressed	Not decompressed
3	no_signal	No signal in channel
4	saturation	Saturation in channel
5	invalid_radiance	Derived radiance outside calibration
6	no_parameters	Calibration parameters unavailable
7	unfilled_pixel	Unfilled pixel

Table 5-5 Exception byte contents and meaning

The exception words for each channel and pixel are initialised at the start of the processing as follows.



exception{ch, view, S, px, t, k} = 0;

Also if for a given view or scan S, if no ISP is present then

exception{ch, view, S, px, t, k} = 1; (ISP_absent)

Step 5.5.1 Calibrate infra red channels

For each scan of each view and for each infra-red channel ch = S7, S8, S9, F1 and F2 and detector element all the pixels of the scan are calibrated as follows.

If $C\{ch, S, px, t, k, view\} > 0$ and
 $(slope_{IR}\{ch, view, S, t, k\} > 100000$ or
 $offset_{IR}\{ch, view, S, t, k\} > 100000)$

then the pixel is valid but there is no valid calibration for pixels of the specified parity on the scan. Set the calibrated pixel value to the `_FillValue` and the corresponding exception value:

$BT_{scene}\{ch, view, S, px, t, k\} = -32768$

exception{ch, view, S, px, t, k} = 64 (no_parameters)

If $C\{ch, view; s, p_v\} \leq 0$ then the channel count value is invalid; set value to `_FillValue` and the corresponding exceptional value

$BT_{scene}\{ch, view, S, px, t, k\} = -32768$

exception{ch, view, S, px, t, k} = 4 (No signal in channel)

If $C\{ch, view; s, p_v\} \geq 65535$ then the channel count value is invalid; set value to `_FillValue` and the corresponding exceptional value

$BT_{scene}\{ch, view, S, px, t, k\} = -32768$

exception{ch, view, S, px, t, k} = 16 (saturation)

Otherwise the pixel count is converted to a radiance

$L_{scene}\{ch, view, S, px, t, k\} = C_{scene}\{ch, view, S, px, t, k\} * slope_{IR}\{ch, view, S, t, k\} + offset_{IR}\{ch, view, S, t, k\}$

eq 5.5-1

Finally, the radiance L_{scene} is converted to BT by linear interpolation in the LUT. The LUT consists of radiance L_{LUT} and brightness temperature BT_{LUT}

If the radiance L_{scene} lies within the range of the `first_value` and `last_value` of the appropriate table relevant to each channel, i.e. if

$first_value(ch) \leq L_{scene}\{ch, view, S, px, t, k\} < last_value(ch)$, eq 5.5-2

find the index g such that



$$L_{LUT}\{q\} \leq L_{scene}\{ch, view, S, px, t, k\} < L_{LUT}\{q+1\}$$

where radiance{g} symbolises the radiance to which the gth tabular brightness temperature corresponds, which is first_value(ch) + g * increment(ch):

$$\begin{aligned} scale &= (L - first_value(ch))/interval(ch) \\ g &= \text{integer part of } (scale) \end{aligned} \tag{eq 5.5-3}$$

Then

$$BT_{scene}\{ch, view, S, px, t, k\} = \text{nearest integer to } (100 * (BT_{LUT}\{ch, g\} + \{BT_{LUT}\{ch, g+1\} - BT_{LUT}\{ch, g\}\} * (scale - g))) \tag{eq 5.5-4}$$

(where the factor of 100 converts the units to 0.01 K).

If the radiance L_{scene} lies outside the range of the table, the calibrated pixel value is set to the Fill Value and the corresponding exceptional value is set:

$$\begin{aligned} BT_{scene}\{ch, view, S, px, t, k\} &= -32768 \\ exception\{ch, view, S, px, t, k\} &= 32 \text{ (Pixel radiance outside calibration)} \end{aligned} \tag{eq 5.5-5}$$

Step 5.5.2 Calibrate fire channels

Fire channels will use the same algorithm as for the TIR channels.

Step 5.5.3 Calibrate visible and SWIR channels to reflectance

In this section the visible and SWIR channels are calibrated into a quantity referred to as 'reflectance' and this is subsequently used in the the cloud test in section 5.5.3.7. It should be noted however, that the term reflectance does not strictly apply to R_{scene} derived below as the cosine of the illumination direction is not corrected for.

For each scan and for each short-wave infrared and visible channel $ch = S1-S6$ all the pixels of the scan are calibrated as follows.

$$\begin{aligned} \text{If } C_{scene}\{ch, view, S, px, t, k\} &> 0 \text{ and} \\ &(\text{slope}_{sol}\{ch, view, S, t, k\} > 100000) \end{aligned} \tag{eq 5.5-6}$$

then the pixel is valid but there is no valid calibration for pixels of the specified parity on the scan. Set the calibrated pixel value to the Fill Value and set the corresponding exceptional value:

$$\begin{aligned} R_{scene}\{ch, view, S, px, t, k\} &= -32768 \\ exception\{ch, view, S, px, t, k\} &= 64 \text{ (no_parameters)} \end{aligned} \tag{eq 5.5-7}$$

If $C\{ch, view, S, px, t, k\} \leq 0$ then the channel count value is invalid; Set the calibrated pixel value to $_FillValue$ and set the corresponding exceptional value:

$$\begin{aligned} R_{scene}\{ch, view, S, px, t, k\} &= -32768 \\ exception\{ch, view, S, px, t, k\} &= 4 \text{ (No signal in channel)} \end{aligned}$$

If $C\{ch, view, s, p_v\} \geq 65535$ then the channel count value is invalid; set value to $_FillValue$ and the corresponding exceptional value



$$R_{scene}\{ch, view, S, px, t, k\} = -32768$$

$$exception\{ch, view, S, px, t, k\} = 16 \text{ (saturation)}$$

eq 5.5-8

Otherwise the pixel count is converted to a reflectance using the slope and average dark counts from the cold black body previously determined.

$$R_{scene}\{ch, view, S, px, t, k\} = (C_{scene}\{ch, view, S, px, t, k\} - \overline{C}_{bbc}\{ch, view, t, k\}) / \text{Gain}(ch) * \text{slope}_{sol}\{ch, view, S, t, k\}$$

eq 5.5-9

If the scene reflectance, R_{scene} lies outside the range of the table, the calibrated pixel value is set to the Fill Value and the corresponding exceptional value is set such that:

$$R_{scene} = -32768$$

$$Exception = 32 \text{ (Pixel radiance outside calibration)}$$

eq 5.5-10

Conversion to top-of-atmosphere radiance is performed after the cloud tests as these are based on reflectance values. The algorithm is described in section 5.5.3.7.

5.5.3.2 Time Domain Averaging of SWIR Channels

The SWIR channels of SLSTR use arrays of 8 detector elements arranged in a matrix of 2 columns by 4 rows, with the long side of the array aligned parallel to the ground track. It follows that the swath is effectively scanned twice in the across-track direction; if the detector columns are designated A and B, then the swath is scanned at 0.5 km nominal resolution by each of the A and B columns independently.

The objective of the present stage is to produce a combined image by adding the two images such that their positions are aligned in the centre of the swath. The resultant combined image will then have a lower signal to noise ratio than the A and B images taken separately.

For each of the TDI (SWIR) channels $ch = 4, 5, 6$ we have a calibrated reflectance

$$R_{scene}\{ch, view, S, p, t, k\}.$$

In order to specify the algorithm formally we need to assume a relationship between the detector index k and the physical arrangement of the detector elements. Strictly such a relationship is forced by the ordering of the pixel counts in the ISP, but this depends on the physical design of the instrument, and is at present TBC.

We shall therefore assume a convention, recognising that this may need to be remapped when the physical layout becomes clear. The relationship adopted is shown in the diagram below, which represents the projection of the pixels on the ground.

	A	B
j=3	3	7
2	2	6
1	1	5
0	0	4
i =	0	1



In the diagram the two-dimensional array of detectors is indexed by i and j , where

$$i = \text{column}(k) = \text{integer part of } [k/4];$$

$$j = \text{row}(k) = k - 4i$$

Therefore

$$k = 4i + j.$$

We assume that the X and Y co-ordinates increase in the directions of increasing column (i) and row (j) index

With these definitions the calibrated reflectances $R\{ch, target; S, p, t, k\}$ can be re-indexed in the form

$$R\{ch, target; S, p, t, i, j\} = R\{ch, target; S, p, t, k = 4i + j\}$$

We can further develop the notation by introducing the new index

$$p' = 2p + t, \quad (0 \leq p' < 2n_t - 1)$$

p' is essentially an index to the PIX05SYNC pulses. we then have

$$R\{ch, target; S, p', i, j\} = R\{ch, target; S, p, t, i, j\}$$

where

$$p' = \text{intpt}(p/2);$$

$$t = 2p - p'$$

For each channel and target we can segregate this into two arrays corresponding to the images sampled by the two columns of detectors A ($i = 0$) and B ($i = 1$)

$$R_A(S, p', j) = R\{ch, target; S, p', 0, j\}$$

$$R_B(S, p', j) = R\{ch, target; S, p', 1, j\}$$

The combined image, which we will denote by R_C , is obtained by adding the two subsets by columns, with the appropriate offset to align the pixels in the centre of the swath. The correct indexing is obtained by considering the X co-ordinates corresponding to each of the pixel positions.

Consider the ground projection of the pixels. Let the X co-ordinate of the centre of the 0.5 km pixel indexed by $\{S, p', k\}$ be $X(p', i)$; we assume that it is essentially independent of scan index S and detector row j in the centre of the swath (although this will not be true elsewhere). In the nadir view, the relationship between the columns of detectors means that, again in the centre of the swath,

$$X(p', 1) = X(p' + 1, 0).$$

The X co-ordinates associated with elements of R_A and R_B are respectively

$$X_A(S, p', j) = X(p', 0)$$

$$X_B(S, p', j) = X(p', 1) = X_A(S, p' + 1, j)$$

$$(0 \leq p' < 2n_t - 1)$$

The arrays must therefore be added so that

$$R_C(S, p'_C, j) = (R_A(S, p'_A, j) + R_B(S, p'_B, j))/2$$

where $p'_B = p'_A - 1$

The algorithm for the nadir view image is thus:

$$R_C(S, p'_C, j) = R_A(S, p'_C, j) \text{ if } p'_C = 0$$



$$R_C(S, p'_C, j) = (R_A(S, p'_C, j) + R_B(S, p'_C - 1, j)) / 2 \text{ if } 1 \leq p'_C \leq 2n_t - 1$$

$$R_C(S, p'_C, j) = R_B(S, p'_C - 1, j) \text{ if } p'_C = 2n_t$$

In order to regrid the arrays R_C , it is necessary to generate the corresponding weighted pixel x and y coordinates. The same equations are used, but with `nadir_x_coordinate` and `nadir_y_coordinate` substituted for the corresponding R .

In the along-track view the relationship between of the detector columns relative to increasing p is reversed. $X(p)$ decreases with increasing p , and the B column trails the A column.. Therefore the relationship between the two columns is

$$X_B(S, p', j) = X(p', 1) = X_A(S, p' - 1, j)$$

The algorithm for the along-track view then becomes

$$R_C(S, p'_C, j) = R_B(S, p'_C, j) \text{ if } p'_C = 0$$

$$R_C(S, p'_C, j) = (R_B(S, p'_C, j) + R_A(S, p'_C - 1, j)) / 2 \text{ if } 1 \leq p'_C \leq 2n_t - 1$$

$$R_C(S, p'_C, j) = R_A(S, p'_C - 1, j) \text{ if } p'_C = 2n_t - 1$$

In order to regrid the arrays R_C , it is necessary to generate the corresponding weighted pixel x and y coordinates. The same equations are used, but with `nadir_x_coordinate` and `nadir_y_coordinate` substituted for the corresponding R .



5.5.3.3 Regrid pixels

The basic concept of this procedure is to remap the measured nadir and oblique instrument pixels from their positions on the curved instrument scans to a uniform grid of points in the common quasi-Cartesian co-ordinate system described in Section 2.4.4. The procedure uses a modified nearest neighbour type of algorithm.

There are two image grids onto which the measurement pixels should be regridded: a 1km grid and a 0.5km grid. Each channel of each view must be regridded separately. Here we outline the approach that is adopted for a single combination of channel and view.

The starting point for a given channel/view is a set of brightness temperature or reflectance values, sampled on an instrument grid which reflects the curved scanning geometry of the SLSTR instrument, together with the X and Y co-ordinates of these instrument pixels in the image co-ordinate system.

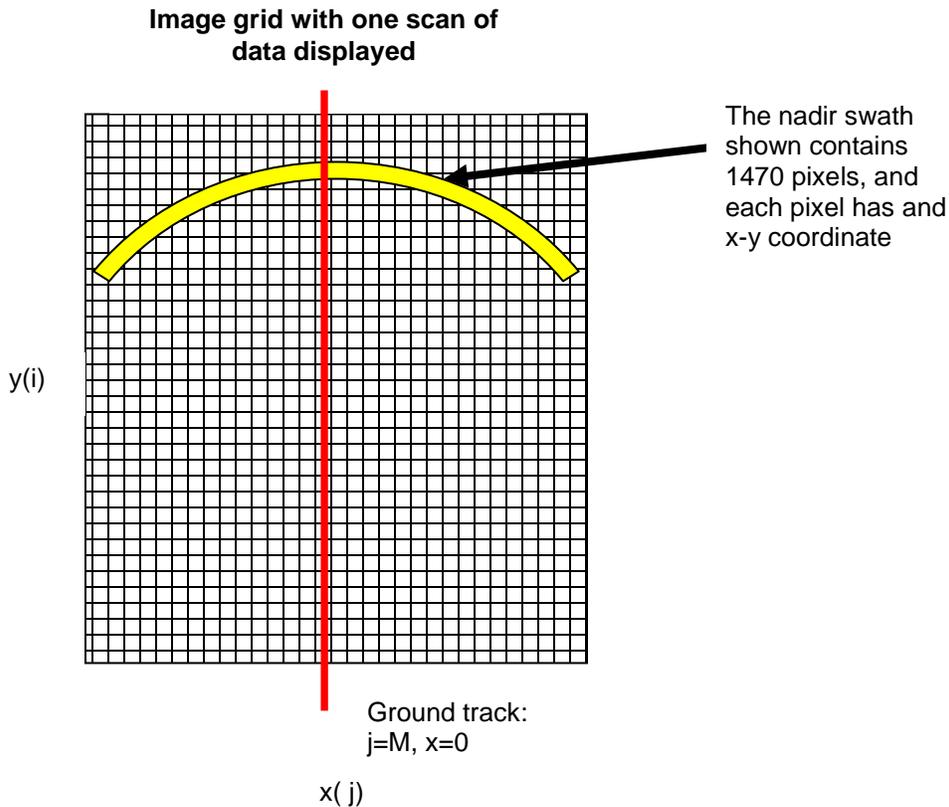


Figure 5-8

The aim of the regridding algorithm is to resample this set of instrument pixels onto a uniform rectangular grid of points in X and Y. Here X is the across-track co-ordinate and Y is the along track co-ordinate.

We adopt the convention that the image grid is indexed by i, j, where the index i is an along-track index (to the rows of the image array) and j is the across-track index (to the columns). This can seem confusing,



because i maps to Y and j to X , but it is consistent with the usual convention for indexing arrays, where $A[i,j]$ represents an array indexed by rows (i) and columns (j).

The image is assumed to comprise N columns, indexed $j = 0$ to $N - 1$. The columns are equally sampled in X at interval dX . so that there is a linear relationship between X and j , which may be written as

$$(1) X(j) = (j - M)dX.$$

where M is a constant such that the column indexed by $j = M$ has $X = 0$ and so corresponds to the nominal ground track, since the definition of X states that $X = 0$ on the ground track. The nominal swath width is NdX . For the thermal channels, $dX = 1$ km, for the solar channels $dX = 0.5$ km.

In the along-track dimension the situation is more complicated, because, following AATSR practice, equal time sampling has been adopted. This means that the sampling interval in Y is given by

$$(2) dY = v dT,$$

where v is the ground track velocity of the satellite, and dT is the appropriate fraction of the scan (SCANSYNC) period. That is, $dT = \text{SCANSYNC} / K$, where K is the number of detector rows for the channel. The ground track velocity of the satellite varies around the orbit, so the spatial sampling interval dY also varies and we cannot assume a linear relationship between Y and the index i . Locally however the relationship is linear to a close approximation. If $Y(i)$ is the Y co-ordinate of the row of samples indexed by i , we can write

$$(3) Y(i + k) = Y(i) + kdY$$

where dY is locally constant, and is approximately equal to dX .

The set of points $[i, j]$ whose co-ordinates are $[X(j), Y(i)]$ for all valid i and j defines the sampling grid. Consider the rectangle whose corners are at the four points $[X(j), Y(i)]$, $[X(j), Y(i+1)]$, $[X(j+1), Y(i+1)]$, $[X(j+1), Y(i)]$; this is a small rectangular area of approximate side 1 km (thermal channels) or 0.5 km (solar channels), and can be taken to define the image pixel at grid point $[j, i]$. The centre of this area is at $[X(j) + 0.5dX, Y(i) + 0.5dY]$, and the set of points $[X(j) + 0.5dX, Y(i) + 0.5dY]$ form the grid of pixel centres, displaced relative to the grid of corner points $[X(j), Y(i)]$ by the vector $[0.5dX, 0.5dY]$.

Relationship between the image grids

As noted above, there are two image grids, at nominal sampling intervals of 1.0 and 0.5 km for the thermal / fire and solar channels respectively. In order to distinguish between the two, let us adopt the notation that the 1.0 km grid is indexed by $[i_{10}, j_{10}]$, and the 0.5 km grid by $[i_{05}, j_{05}]$. Note that as before i is the along-track and j is the across-track index.

The 1.0 km image grid therefore consists of the set of points having co-ordinates

$$X[i_{10}, j_{10}], Y[i_{10}, j_{10}]$$

where

$$X[i_{10}, j_{10}] = (j_{10} - M)dX$$

and

$$Y[i_{10} + 1, j_{10}] = Y[i_{10}, j_{10}] + dY$$

The 0.5 km grid consists of the set of points having co-ordinates

$$X[i_{05}, j_{05}], Y[i_{05}, j_{05}]$$

where

$$X[i_{05}, j_{05}] = (j_{05} - M)(dX / 2)$$

and

$$Y[i_{05} = 2i_{10}, j_{05}] = Y[i_{10}, j_{10}]$$



In other words the points of the 1.0 km image grid having the indices $[i_{10}, j_{10}]$ are coincident with the points of the 0.5 km grid having indices

$$i_{05} = 2i_{10}; j_{05} = 2j_{10}$$

We regard the grid points as defining the lower left corners (not the centres) of the image pixels. (In other words the geometrical instrument pixel $[i_{10}, j_{10}]$ at 1 km resolution includes the four 0.5 km pixels $[2*i_{10}, 2*j_{10}]$, $[2*i_{10}+1, 2*j_{10}]$, $[2*i_{10}, 2*j_{10}+1]$, $[2*i_{10}+1, 2*j_{10}+1]$.)

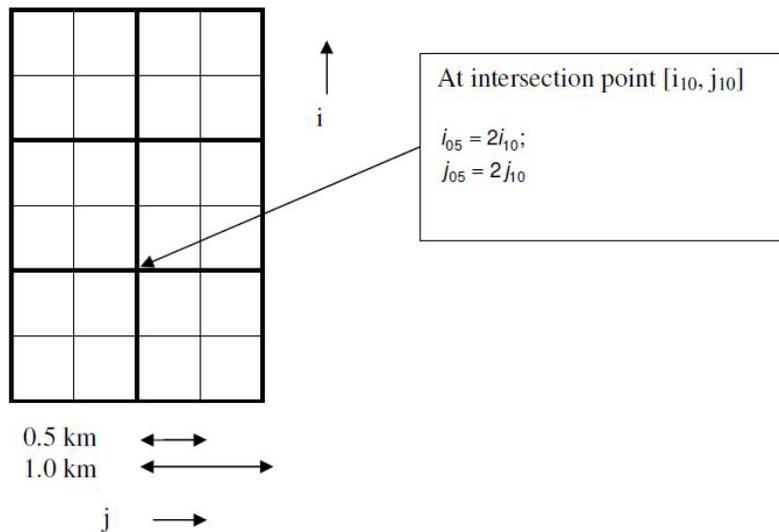


Figure 5-9

Prior processing will have generated input arrays on 4 different instrument grids for each of the two views; the 1 km grid for the 3 thermal and fire channels, 0.5 km A and B grids for the solar channels, and a combined 0.5 km grid for the TDI channels.

The four instrument grids are as follows:

- 1) In terms of the notation defined in Section 4.2.3, the thermal and fire channel instrument grid is represented by the set of indices $S, p, k = 0, 1$, representing scan, pixel and detector number. Alternatively they may be indexed by scan trace and pixel index s, p , which is the approach adopted in Section 5.5.3.3.
- 2) The instrument grid for the A stripe (VIS channels and SWIR) is represented by the index set $S, p, t=0, 1, k = 0, \dots, 3$.
- 3) The instrument grid for the B stripe (SWIR only) by $S, p, t=0, 1, k = 4, \dots, 7$ (assuming the detector indexing illustrated in Section 5.5.3.2 (page 118). Again the ATBD adopts the combined indices $s = 4S + \text{row}(k)$, $p' = 2p + t$. (The prime is then dropped for notational convenience).
- 4) The TDI grid is that defined in Section 5.5.3.2.



Each channel must be regridded into one of two output grids (at 1 km and 0.5 km resolution, as appropriate). There are two output grids for each view, which differ in that the grid for the solar channels has twice the dimensions of that for the solar thermal channels. The indices i and j refer to the appropriate array, as do the MAX and MIN limits. The 1.0 and 0.5 km image grids are coincident in the sense that alternate samples of the 0.5 km grid coincide with the samples of the 1.0 km grid. In detail, for each view we will have the following:

- Three thermal and two fire channels, to be resampled from the 1 km instrument grid to the 1 km image grid.
- Six 'A stripe' solar channels to be regridded from the 'A stripe' instrument grid to the 0.5 km image grid (assuming that the visible channel detectors correspond to the A stripe).
- Three 'B stripe' SWIR channels to be regridded from the 'B stripe' instrument grid to the 0.5 km image grid (again assuming that the visible channel detectors correspond to the A stripe).
- Three combined TDI SWIR channels to be regridded from the TDI instrument grid to the 0.5 km image grid.

In order to retain the link between the regridded pixels and their original positions in the measurement geometry, the original instrument scan, pixel and detector number are retained in an ADS.

Step 5.5.1 Initialize Image Arrays

Initialize the regridded data arrays. This is necessary to ensure (a) that the requirement for cosmetic fill can be recognised; (b) that unfilled pixels can be identified at the conclusion of the regridding and cosmetic fill processes; and (c) that the regridded maximum and minimum temperature variables are initialized.

```
nadir_fill_state(i, j) = UNFILLED_PIXEL
oblique_fill_state(i, j) = UNFILLED_PIXEL
```

These are the arrays which will contain the fill state of each pixel after regridding and cosmetic fill. The elements of these arrays may take one of 3 possible values:

```
NATURAL_PIXEL The pixel has been assigned by the regridding algorithm
COSMETIC_PIXEL The pixel value has been assigned by the cosmetic fill algorithm
UNFILLED_PIXEL Neither the regridding nor cosmetic fill algorithm have assigned a value to this pixel
```

Unfilled pixels occur because the cosmetic fill algorithm requires that an initially unfilled pixel can only be filled by an adjacent pixel. If a pixel has no naturally filled neighbours, it will remain unfilled. The pixel fill state will eventually appear in the product as an exception flag.

The fill values are used in the global confidence words in table of SY4 and the thermal and visible MDS exception words (see tables below).

Bit Number	Text Code	Description
0	ISP_absent	ISP absent
1	pixel_absent	Pixel absent
2	not_decompressed	Not decompressed
3	no_signal	No signal in channel
4	saturation	Saturation in channel
5	invalid_radiance	Derived radiance outside calibration
6	no_parameters	Calibration parameters unavailable
7	unfilled_pixel	Unfilled pixel



Bit	Text code	Meaning if set	Comment
0	coastline	coastline in field of view	
1	ocean	ocean in field of view	
2	tidal	tidal zone in field of view	
3	land	land in field of view	
4	inland_water	inland water in field of view	
5		(spare)	
6		(spare)	
7		(spare)	
8	cosmetic	cosmetic fill pixel	
9	duplicate	duplicate pixel not regridded	
10	day	pixel in daylight	
11	twilight	pixel in twilight	
12	sun_glint	sun glint in pixel	
13	snow	snow	
14	summary_cloud	summary cloud test	
15	summary_pointing	summary pointing	

$nadir_min_anc_temps(i, k) = +999.0$ ($k = 0, 1, \dots, 5$)
 $nadir_max_anc_temps(i, k) = -999.0$ ($k = 0, 1, \dots, 5$)

$oblique_min_anc_temps(i, k) = +999.0$ ($k = 0, 1, \dots, 5$)
 $oblique_max_anc_temps(i, k) = -999.0$ ($k = 0, 1, \dots, 5$)

The above arrays hold the maximum and minimum detector temperatures recorded on the instrument scans that contribute to a given image row.

$nadir_packet_invalid(i) = 0$
 $oblique_packet_invalid(i) = 0$

Invalidity flags for the source packets contributing to each pixel row.

Also initialise the instrument scan and pixel number to zero for all values of i, j :

$nadir_scan_number(i, j) = 0$
 $nadir_pixel_number(i, j) = 0$
 $oblique_scan_number(i, j) = 0$
 $oblique_pixel_number(i, j) = 0$

These arrays are for use by level 1c and 2 processing to show the origin on the instrument grid of each pixel.

For each MDS initialise the exception flags for each channel in each view to FALSE:

$exception(i, j) = FALSE$;
 $exception_orphan(i, j) = FALSE$

eq 5.5-11

Step 5.5.2 Regrid Image Pixels, Nadir View

The following steps are implemented for each instrument pixel in the nadir view.



As in section 5.3.3.1, instrument pixel will be identified by the following indices

- s Scan number;
- p'_n Relative pixel number, nadir scan ($0 \leq p'_n < \text{MAX_NADIR_PIXELS}$)
- p'_a Relative pixel number, oblique (inclined) scan ($0 \leq p'_a < \text{MAX_ALONG_TRACK_PIXELS}$)
- k Detector element index. $k = 0, 1$ for the thermal channels, $0, \dots, 3$ for the solar channels (the range of k is reduced following TDI).

In Section 5.3.3.7 the x and y co-ordinates of each instrument pixel were determined. We introduce the notation $nadir_x_coord(s, p'_n, k)$, $nadir_y_coord(s, p'_n, k)$ for the x and y co-ordinates respectively of the nadir pixel identified by indices, s, p'_n, k . We also introduce MIN_NADIR_X , MAX_NADIR_X to denote the lower and upper limits of the instrument nadir swath in x units. Finally we define a factor $scale$, which takes the value 1 for the thermal channels and 2 for the solar channels (it is the reciprocal of the nominal resolution, in km.) The target first acquisition ($FIRST_<VIEW>_PIXEL_NUMBER$) from the instrument ISPs. This is incremented every $PIX10SYNC$, even for the solar and SWIR channels. The first pixel number of the solar and SWIR channels will therefore be twice that of the $FIRST_<VIEW>_PIXEL_NUMBER$ provided in the ISP, and therefore the $scale$ factor is used to account for this in the algorithms.

For each source packet scan s the channels are processed as follows. For each nadir pixel $p'_n = 0, (\text{MAX_NADIR_PIXELS} - 1)$, for all applicable values of k , in the nadir scan s for which

$$\begin{aligned} &nadir_x_coord(s, p'_n, k) \geq MIN_NADIR_X \text{ and} \\ &nadir_x_coord(s, p'_n, k) < MAX_NADIR_X, \end{aligned} \tag{eq 5.5-12}$$

the regridding indices are calculated. Find the index ig such that

$$track_y(ig + K) \leq nadir_y_coord(s, p'_n, k) < track_y(ig + K + 1)$$

($track_y, K$: Section 5.3.3.3.) Then

$$\Delta y = (track_y(ig + K + 1) - track_y(ig + K)) / NGRANULE \tag{eq 5.5-13}$$

$$\delta y = scale(nadir_y_coord(s, p'_n, k) - track_y(ig + K)) / \Delta y \tag{eq 5.5-14}$$

$$i' = \text{integer part of } (\delta y) \tag{eq 5.5-15}$$

$$i = scale \cdot NGRANULE \cdot ig + i' \tag{eq 5.5-16}$$

$$\delta x = scale(nadir_x_coord(s, p'_n, k) - MIN_NADIR_X) \tag{eq 5.5-17}$$

$$j = \text{FIX}(\delta x) \tag{eq 5.5-18}$$

If i is negative, the pixel is outside the image bounds. (This should not happen, provided the image bounds are properly chosen, but if it does) set j to the next available orphan pixel index and continue, at **Step 5.5.2.4**.

The absolute pixel number of the pixel under consideration is

$$p_n = p'_n + (scale * FIRST_NADIR_PIXEL_NUMBER) \tag{eq 5.5-19}$$

The instrument scan number is

$$s_n = 2 \cdot scale \cdot (s + \sigma) + k \tag{eq 5.5-20}$$

where σ is the additive constant introduced earlier. If

$$s_n = nadir_scan_number(i - i', j) + i' \tag{eq 5.5-21}$$



and

$$p_n = \text{nadir_pixel_number}(i - i', j) \quad \text{eq 5.5-22}$$

Step 5.5.2.4 Regrid nadir pixels

Note that these steps should only be executed if the value of *nadir_fill_state*[*i*, *j*] has the value *UNFILLED_PIXEL*. Otherwise the pixel and its co-ordinates should be saved as an orphan Pixel.

Save the instrument scan and pixel number for the regridded image point in the nadir view:

$$\text{nadir_scan_number}(i, j) = s_n \quad \text{eq 5.5-23}$$

$$\text{nadir_pixel_number}(i, j) = p_n = p'_n + (\text{scale} * \text{FIRST_NADIR_PIXEL_NUMBER}) \quad \text{eq 5.5-24}$$

Finally the calibrated nadir channel data and exception flags for each channel are copied to the regridded data arrays. In the following, the index *ch* refers to the channel. *I* refers to the calibrated intensity (Temperature or reflectance, as appropriate) of the destination channel. *BT* and *R* refer to the calibrated brightness temperature or radiance, as appropriate, of the instrument pixel channels.

For the thermal infra-red channels *ch* = S7-S9, F1-F2

$$I(\text{ch}, \text{nadir}; i, j) = \text{BT}(\text{ch}, \text{nadir}; s, p_n) \quad \text{eq 5.5-25}$$

and for the remaining (solar) channels

$$I(\text{ch}, \text{nadir}, i, j) = \text{R}(\text{ch}, \text{nadir}; s, p_n).$$

$$\text{exception_flag}(\text{ch}, \text{nadir}, i, j) = \text{Exception}(\text{ch}, \text{nadir}, s, p_n) \quad \text{eq 5.5-26}$$

Similarly other parameters computed at full resolution (e.g. solar and view angles, *x* & *y*) should also be regridded in the same way. Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled using bilinear interpolation in *x*, *y* from the full resolution grid on the Tie Point grid for the product after the cosmetic fill has been performed. An internal variable containing the SZA, SAA at full resolution should be kept to be used at cloud flagging.

Finally the offset of the source pixel from the corner of the regridded pixel is saved for use by the cosmetic fill routine.

$$x_offset(\text{nadir}, i, j) = \text{fractional part of } \delta x \quad \text{eq 5.5-27}$$

$$y_offset(\text{nadir}, i, j) = \text{fractional part of } \delta y \quad \text{eq 5.5-28}$$

$$\text{fill_state}(\text{nadir}, i, j) = \text{NATURAL_PIXEL} \quad \text{eq 5.5-29}$$

Step 5.5.3 Regrid Image Pixels, Oblique View

This process is repeated for the oblique scan as follows.

We introduce the notation *oblique_x_coord*(*s*, *p'*_{*a*}, *k*), *oblique_y_coord*(*s*, *p'*_{*a*}, *k*) for the *x* and *y* co-ordinates respectively of the oblique pixel identified by indices, *s*, *p'*_{*a*}, *k*. We also introduce *MIN_OBLIQUE_X*, *MAX_OBLIQUE_X* to denote the lower and upper limits of the instrument oblique swath in *x* units (provisionally we expect *MIN_OBLIQUE_X* = -384 km, *MAX_OBLIQUE_X* = +384 km (TBC).)

For each oblique pixel *p'*_{*a*} = 0, (*MAX_OBLIQUE_PIXELS* - 1) for all applicable values of *k* in the oblique scan *s* with

$$\begin{aligned} \text{oblique_x_coord}(s, p'_a, k) &\geq \text{MIN_OBLIQUE_X} \text{ and} \\ \text{oblique_x_coord}(s, p'_a, k) &< \text{MAX_OBLIQUE_X} \end{aligned}$$

$$\text{eq 5.5-30}$$



the regridding process proceeds exactly as before, except that references to oblique view quantities replace those of the equivalent nadir view variables throughout.

The index ig such that

$$track_y(ig + K) \leq oblique_y_coord(s, p'_a, k) < track_y(ig + K + 1) \quad \text{eq 5.5-31}$$

is found. Then

$$\Delta y = (track_y(ig + K + 1) - track_y(ig + K)) / NGRANULE \quad \text{eq 5.5-32}$$

$$\delta y = scale(oblique_y_coords(s, p'_a, k) - track_y(ig + K)) / \Delta y \quad \text{eq 5.5-33}$$

$$i' = \text{intpt}(\delta y) \quad \text{eq 5.5-34}$$

$$i = scale \cdot NGRANULE \cdot ig + i' \quad \text{eq 5.5-35}$$

$$\delta x = scale(oblique_x_coords(s, p'_a, k) - MIN_OBLIQUE_X) \quad \text{eq 5.5-36}$$

$$j = \text{FIX}(\delta x) \quad \text{eq 5.5-37}$$

If i is negative, the pixel is outside the image bounds. (As in the nadir case, this should not happen, provided the image bounds are properly chosen, but if it does) set j to the next available orphan pixel index and continue, at **Step 5.5.3.4**.

If $i' = 0$ (that is, if the value of i corresponds to a granule row), or this is the first scan to be regridded, or
 $oblique_scan_number(i_g, j) = 0$

The absolute pixel number of the pixel under consideration is

$$p_a = p'_a + (scale * FIRST_OBLIQUE_PIXEL_NUMBER) \quad \text{eq 5.5-38}$$

The instrument scan number is

$$s_a = 2 \cdot scale \cdot (s + \sigma) + k \quad \text{eq 5.5-39}$$

where σ is the additive constant introduced earlier. If

$$s_a = oblique_scan_number(i - i', j) + i'$$

and

$$p_a = oblique_pixel_number(i - i', j)$$

then go to **Step 5.5.3.4**.

eq 5.5-40

Step 5.5.3.4 Regrid oblique pixels

Note that these steps should only be executed if the value of $oblique_fill_state[i, j]$ has the value *UNFILLED_PIXEL*. Otherwise the pixel and its co-ordinates should be saved as an orphan Pixel.

Save the scan and pixel number for the regridded image point in the oblique view:

$$oblique_scan_number(i, j) = s_a \quad \text{eq 5.5-41}$$

$$oblique_pixel_number(i, j) = p_a = p'_a + (scale * FIRST_OBLIQUE_PIXEL_NUMBER) \quad \text{eq 5.5-42}$$



Finally the calibrated oblique view data for each channel are copied to the regridded data arrays.

For the infra-red channels $ch = S7-S9, F1-F2$

$$I(ch, oblique; i, j) = BT(ch, oblique; s, p_i)$$

and for the visible channels $ch = S1-S6$

$$I(ch, oblique, i, j) = R(ch, oblique; s, p_i).$$

$$exception_flag(ch, oblique, i, j) = exception(ch, oblique, s, p_n) \quad \text{eq 5.5-43}$$

Finally the offset of the source pixel from the corner of the regridded pixel saved for use by the cosmetic fill routine.

$$x_offset(oblique, i, j) = \text{fractional part of } \delta x \quad \text{eq 5.5-44}$$

$$y_offset(oblique, i, j) = \text{fractional part of } \delta y \quad \text{eq 5.5-45}$$

Set the fill state for the pixel just regridded:

$$fill_state(oblique, i, j) = NATURAL_PIXEL \quad \text{eq 5.5-46}$$

Similarly other parameters computed at full resolution (e.g. solar and view angles, x & y) should also be regridded in the same way. Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled on the Tie Point grid for the product after the cosmetic fill has been performed. An internal variable containing the SZA, SAA at full resolution should be kept to be used at cloud flagging.

Step 5.5.4 Regrid Auxiliary Information

Other nadir view information from the source packet is now regridded. This section is based on the functions performed during AATSR processing, and may require modification for the SLSTR instrument following instrument testing.

To regrid the remaining information find the first and last regridded image rows to which the instrument scan contributes. The extreme scan y positions are :-

$$nadir_y_coord(s, 0) \quad \text{eq 5.5-47}$$

$$nadir_y_coord(s, number_of_scan_pixels - 1) \quad \text{eq 5.5-48}$$

$$nadir_y_coord(s, number_of_scan_pixels / 2) \quad \text{eq 5.5-49}$$

Find the minimum and maximum of these positions $ymin$ and $ymax$. Then find the regridding indices $imin$ and $imax$ using the method for calculating i detailed above. For each scan $i = imin, imax$ do the following: Test $packet_error(s)$ in turn for the following.

Raw Packet fails basic validation error

Checksum errors

Scan and Flip Mirror Errors

If equality is found the corresponding bit in the global ADS is set.

Next the minimum and maximum values of each of the 6 ancillary temperatures for the current scan are updated when the current values are exceeded by the instrument scan values.

ie for $janc = 0$ to 5

if $anc_unconv(s, janc) > 0$ then

if $anc_temp(s, janc) < nadir_min_anc_temps(i, janc)$ then

$$nadir_min_anc_temp(i, janc) = anc_temp(s, janc)$$

if $anc_temp(s, janc) > nadir_max_anc_temps(i, janc)$ then

$$nadir_max_anc_temp(i, janc) = anc_temp(s, janc)$$

end if

$$\quad \text{eq 5.5-50}$$



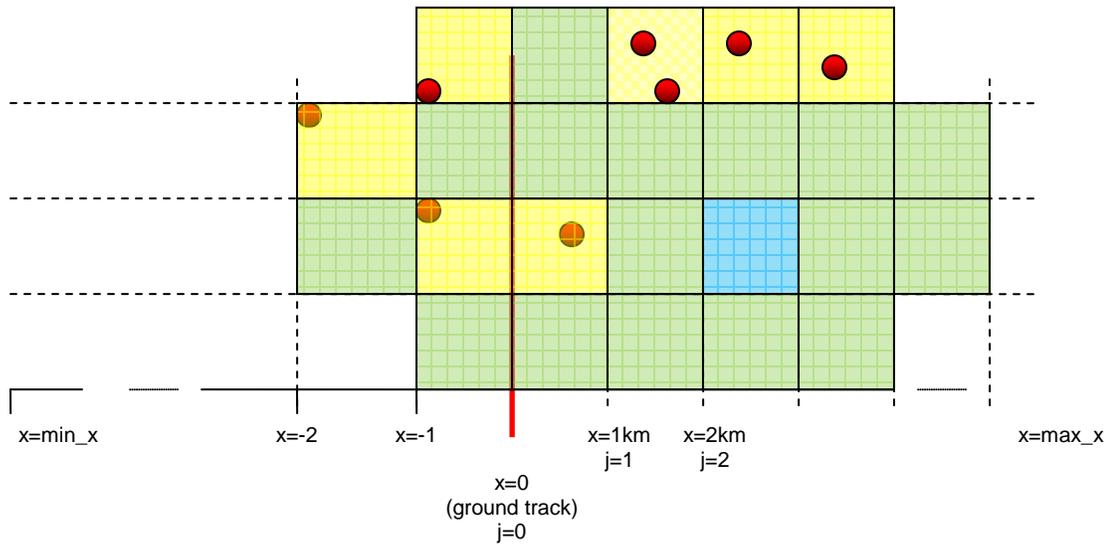
This process is repeated to obtain the oblique scan information.

5.5.3.4 Cosmetic Fill

This process is carried out on the regrided nadir image (view = nadir) and then on the regrided oblique image (view = oblique), and is illustrated by the figure 5-10 below:



Example of cosmetic fill for 1km image grid.



● Circles represent the x and y coordinates of the centre of the measurement pixels, x_n , y_n as previously defined

 Image grid square filled with measurement grid data or exception flag. **Naturally filled.**

 Image grid square filled with measurement grid data point from the nearest neighbour. **Cosmetically filled.**

 Image grid square not filled. **Unfilled pixel.**

 Image grid square Naturally filled but there are two measurement pixels contained in the same square. One of these is an **orphan pixel.**

Figure 5-10

Step 5.5.1 Fill Image pixels

For each pixel (i, j) in the regridded image check the fill state flag and if the pixel is unfilled, ie $\text{fill_state}(\text{view}, i, j) = \text{UNFILLED_PIXEL}$,

eq 5.5-51



an attempt is made to fill it using data from an adjacent pixel as follows.

Examine the 8 regridded pixels adjacent to the unfilled pixel. Each candidate pixel (i_c, j_c) must satisfy the following tests:

1. It is not outside the limits of the swath;
 2. It is a naturally filled pixel
- $$\text{fill_state}(\text{view}, i_c, j_c) = \text{NATURAL_PIXEL}$$

eq 5.5-52

If no candidate pixels are found then the pixel remains unfilled and the pixel values in every channel are set to the PIXEL_UNFILLED value.

$$l(\text{ch}, v, i, j) = \text{PIXEL_UNFILLED}$$

Set the corresponding exception flag to unfilled pixel accordingly:

$$\text{exception_flag}(\text{ch}, \text{view}, i, j) = 128 \text{ (unfilled pixel)}$$

eq 5.5-53

If candidate pixels are found then for each candidate the distance d of the source pixel from the centre of the unfilled target pixel is calculated.

$$dx = j_c - j + x_offset(\text{view}, i_c, j_c) - 0.5.$$

eq 5.5-54

$$dy = i_c - i + y_offset(\text{view}, i_c, j_c) - 0.5.$$

eq 5.5-55

$$d = \sqrt{dx^2 + dy^2}$$

eq 5.5-56

The candidate with the smallest distance is selected.

The fill state flag for the target pixel is now set to indicate a cosmetically filled pixel and the pixel values from the selected candidate pixel are copied to the target pixel for all channels ch .

$$\text{fill_state}(\text{view}, i, j) = \text{COSMETIC_PIXEL}.$$

eq 5.5-57

$$l(\text{ch}, v, i, j) = l(\text{ch}, v, i_c, j_c)$$

eq 5.5-58

Set the corresponding exception byte value accordingly:

$$\text{exception_flag}(\text{ch}, \text{view}, i, j) = 2 \text{ (cosmetic fill)}$$

eq 5.5-59

Step 5.5.2 Fill Auxiliary Information

Any gaps in the band edge solar angles and band centre solar angles must now be filled. A gap is filled using data from an adjacent valid point along track in the arrays, or if this fails, an adjacent valid point across track is used.

Step 5.5.3 Fill Nadir Scan Number Array

In order to permit the array of nadir view instrument scan numbers ($\text{scn_nadir}(ig, j)$,) to be used during Level 2 processing to time tag averaged product cells, it is also 'cosmetically filled' in a similar way.

```

for all ig
  for j = 1, 511
    if scn_nadir(ig, j) = 0 then
      scn_nadir(ig, j) = scn_nadir(ig, j - 1)
    end if
  end for
  for j = 510, 0

```



```
        if scn_nadir(ig, j) = 0 then
            scn_nadir(ig, j) = scn_nadir(ig, j + 1)
        end if
    end for
end for
```

eq 5.5-60



5.5.3.5 Surface classification

The pixel classification is represented as a small number of classification flags, one for each distinct surface type. The classification is performed as follows:

- Calculate the optical ray corresponding to the centre of the pixel FOV.
- Calculate the intersection of the ray with the topographic (TBC) surface of the earth.
- Look up the value of the surface classification map cell containing the intersection point.
- Set classification flags for all surface types detected at the centre of the FOV.
- If several pixel radiances are combined as part of a time domain integration (TDI) algorithm, then the classification flags for the composite pixel are set to the bitwise OR of the flags in the contributing pixels.
- The result of the classification test is promoted to L1b by the same nearest-neighbour algorithm that is applied to the corresponding pixel radiance.

The surface classification flags are listed in Table 5-6 below and written to the global flags ADS.

Table 5-6: Surface Classification Flags

Bit	Text code	Meaning if set	Comment
0	coastline	coastline in field of view	
1	ocean	ocean in field of view	
2	tidal	tidal zone in field of view	
3	land	land in field of view	
4	inland_water	inland water in field of view	

5.5.3.5.1 Commentary

No flags will be set for pixels for which do not contain any re-gridded radiances. If required, a default value representing the centre of the L1b pixel can be substituted (TBD) but, as these pixels cannot be used in subsequent processing, the more obvious approach would be to leave the flags unset, indicating an exception condition.

Separate classifications must be derived for each instrument view, resolution and (for channels S4 – S6) both the “a” and “b” detector “stripes”.

Higher-level products which use combinations of L1b products may need to interpret several classification words. In most cases the correct composite surface classification flags will be the bitwise OR of all contributing pixel surface classification flags.



5.5.3.6 Identification of Cloud Affected Pixels

The identification of cloud affected pixels is accomplished by applying in turn a series of tests to the brightness temperature data in the 12, 11 and 3.7 micron channels, and to the reflectance data in the short-wave infra-red visible channels. The pixel is flagged as cloudy if any one of the tests indicates the presence of cloud. Table 5-7 below summarises the cloud clearing tests to be applied. All of the tests are of course conditional on the appropriate infra-red or visible channel data being available. The theoretical basis of the majority of these tests is set out in Reference [RD-7]. Most of the tests listed have AATSR heritage, but two are new; these are the 1.375 micron threshold test and the 2.25 micron histogram test, which utilize the new short-wave infra-red channels of SLSTR.

Table 5-7: Cloud Clearing Tests

Test	Comments
gross cloud test	applied to nadir and oblique views separately
thin cirrus test	applied to nadir and oblique views separately
medium/high level cloud test	applied to nadir and oblique views separately
fog/low stratus test	applied to nadir and oblique views separately
11 micron spatial coherence test	applied to nadir and oblique views separately
1.6 micron histogram test	applied to nadir and oblique views separately
11/12 micron nadir/oblique test	uses both views
11/3.7 micron nadir/oblique test	uses both views
infra-red histogram test	applied to nadir and oblique views separately
Visible channel cloud test	applied to nadir and oblique views separately
1.375 micron threshold test	applied to nadir and oblique views separately
2.25 micron histogram test	applied to nadir and oblique views separately
Snow-covered surface test	applied to nadir and oblique views separately

Some of the tests depend on the results from the tests performed previously and hence the order in which they are applied is important. The infrared histogram test is applied after the 1.6 micron and thermal infra-red tests, and only uses those pixels that have not been flagged as cloudy by any of the preceding tests. The 1.6 micron test operates only on pixels not previously flagged as cloudy by the gross cloud test or the thin cirrus and 11 micron spatial coherence tests (the other single view tests that operate on daytime data), and must therefore follow these. The large-scale component of the 11 micron spatial coherence test operates only on pixels that have not been flagged by the gross cloud, thin cirrus, medium/high level and fog/low stratus tests, and should therefore follow them.

The visible channel cloud test is applied independently of the other tests, and follows the infra-red histogram test; since the visible channel test is currently only applied over land, while the infra-red histogram test is applied only to sea pixels, there is no point in introducing a dependency of the latter on the former, and the visible channel test is therefore applied after the infra-red histogram test.

The two tests using the two new short-wave channels at 1.375 and 2.25 microns are considered untried, and it is therefore not considered prudent to make any of the other tests dependent on them at this stage, although the 2.25 micron test can have the same dependency on other daytime tests as the 1.6 micron test. These therefore follow the other tests.

The snow test is not considered to be a cloud test as such, owing to the difficulty of distinguishing snow- or ice covered surfaces from ice cloud, and it does not contribute to the overall cloud flag. It is therefore applied last.



The tests should be applied in the order in which they appear in Table 5-7. In the following the tests are described in the order in which they are applied, except that the 11 micron spatial coherence test (Section 5.5.3.6.1) appears out of sequence, and should be applied after the fog/low stratus test (Section 5.5.3.6.5) for the reason stated.

The 1.6 micron test and the visible channel test operate on daytime data only. The tests involving the 3.7 micron channel, on the other hand, are only applied to night-time data, because reflected solar radiation may be significant in this channel during the day. Those tests that involve the 11 and 12 micron channels are applicable to both daytime and night-time data. Not all of the tests are implemented over land. Each test makes use of a look-up table of parameters with which the brightness temperature or reflectance data is compared. Where tests are applied to oblique and nadir view images separately, the parameters may be defined separately for the two cases. More generally, the comparison parameters may depend on the air mass in the line of sight, and this is implemented by allowing the tabular parameters to depend upon the across-track position.

The cloud tests are performed at the native resolution of the detector elements – i.e. for TIR detectors cloud flags are computed at 1km resolution and VIS/SWIR at 0.5km resolution. In detail, the cloud tests are all based on the regridded brightness temperatures or reflectances, which are sampled on a rectangular grid in the cartesian co-ordinates X and Y, where X is the across-track co-ordinate and Y is the along track co-ordinate. For each view there are two grids, at 1 km and 0.5 km resolution, as appropriate, so the grid for the solar channels has twice the dimensions of that for the solar thermal channels. The 1.0 and 0.5 km image grids are coincident in the sense that alternate samples of the 0.5 km grid coincide with the samples of the 1.0 km grid.

We adopt the convention that the grid is indexed by i, j , where the index i is an along-track index (to the rows of the image array) and j is the across-track index (to the columns). With this convention, suppose the thermal image arrays have N cols, indexed $j_{05} = 0$ to $N - 1$. Then the solar channel arrays will have $2N$ columns indexed $j_{05} = 0, 2N-1$. Similarly we can index the arrays in the along-track direction with i_{10} and i_{05} for the 1.0 km and 0.5 km channels respectively.

In those cases where a test based on the solar channels uses thermal channel data, the thermal channel value associated with the solar channel pixel at $[i_{05}, j_{05}]$ is that indexed by

$$\begin{aligned} i_{10} &= \text{integer part of } (i_{05}/2) \\ j_{10} &= \text{integer part of } (j_{05}/2) \end{aligned}$$

(In other words the geometrical instrument pixel $[i_{10}, j_{10}]$ at 1 km resolution includes the four 0.5 km pixels $[2*i_{10}, 2*j_{10}]$, $[2*i_{10}+1, 2*j_{10}]$, $[2*i_{10}, 2*j_{10}+1]$, $[2*i_{10}+1, 2*j_{10}+1]$). The 0.5km grid replicates the 1km cloud tests for each pixel.

In any case where a test based on the thermal channels uses solar channel data, the Logical OR of the 0.5km flags within that 1km pixel are used. Note there is no interpolation.

In general we expect the AATSR cloud tests to be carried over largely unchanged, apart from some changes to the definitions of the processed sub-areas associated with the histogram and spatial coherence tests to account for the wider swaths of SLSTR compared to AATSR. In the particular case of the oblique view, the AATSR forward view tests should carry over essentially unchanged, since these pixels are measured at an essentially constant air mass or (i.e. zenith angle).

In the case of the nadir view, however, the range of air mass values covered by the nadir scan is much wider than with AATSR. Where the auxiliary constants or look-up tables used by the tests depend on air mass in the line of sight, it may therefore be necessary to divide the nadir swath into across-track ranges within which different constants or tables apply. In the central region of the swath (say between -256 and +256 km across-track), the AATSR nadir view auxiliary data sets should apply with only minor changes. At the swath edge, towards -900 km, where the observational incidence angle approaches 55 degrees,



the AATSR forward view tables may be more appropriate, while between these ranges an intermediate set of values may be required.

More detailed notes are given under the heading of each test below.

The availability of the new channels at 2.25 and 1.375 microns will permit the implementation of new tests. In the following we propose single channel tests based on the 1.6 micron test but using each of the new channels separately.

For the SWIR channels (S4, S5 and S6) the cloud tests can be applied to either A, B or C, the average normalised radiance (reflectance) values. This could be an option in the GPP to select which value to use with the default to be the average pixels.

5.5.3.6.1 *Spatial coherence test (11 micron)*

The test uses a look-up table containing the following parameters:

SEA_MAX_DEV
LAND_DAY_MAX_DEV
LAND_NIGHT_MAX_DEV
COHERENCE_RESET_THRESH
COH_AREA_SIZE_X
COH_AREA_SIZE_Y
COH_FRACTION_PASSED
COH_ADJ_THRESH_LAND
COH_ADJ_DIF_LAND
CLOUDY_BOX_THRESH
X1
X2

These are not view dependent. The view dependent parameters are:

COH_AREA_DIF
COH_MIN_DIF
COH_AREA_THRESH

with the suffix of either `_INT`, `_NA` or `_OB` to indicate whether it is used with intermediate, nadir or oblique views respectively.

Modifications to the baseline AATSR test: This test actually comprises two tests; the small-scale and the large-scale spatial coherence tests.

In the case of the small-scale test there are no major changes (except that the final overlapping 3 by 3 group is not needed if the swath widths are defined to be multiples of three pixels).

In the case of the large-scale test, there are three view-dependent (i.e. air mass dependent) parameters as identified above. We expect to divide the nadir swath into zones by across-track limits X_1 , X_2 such that:

if $-X_1 < X < X_1$ the AATSR nadir view constants are used;
if $X_1 < \text{abs}(X) < X_2$, a new intermediate set of values are used;
if $X_2 < \text{abs}(X)$, the AATSR forward view constants are used.

The AATSR test tiles the image with sub-areas of `COH_AREA_SIZE` pixels square, where in practice `COH_AREA_SIZE = 128`. Currently the width of the oblique view is defined to be a multiple of 128 pixels, so this approach can be carried over unchanged. However in the case of the nadir view the definition is not a multiple of 128. It may be appropriate to distinguish between the oblique dimension `COH_AREA_SIZE_Y = 128` and the across-track dimension `COH_AREA_SIZE_X (= 105 pixels (TBC))`.

The test is performed separately for the nadir and oblique views. The nadir view 12 μm and 11 μm BT data are used in the nadir view test and the oblique view 12 μm and 11 μm BT data are used in the oblique view test. In the following, it is assumed that the size of the nadir and oblique swaths are N km, where N will differ for each viewing direction. The across track co-ordinate is x and is indexed by j , the along track co-



The large scale spatial coherence test uses data from sub-areas of COH_AREA_SIZE_X by COH_AREA_SIZE_Y and is described in steps 5.5.3.6.1-8 to 9. (N.B. The value of COH_AREA_SIZE must be chosen so that it is one of the factors of 512. Its current value is 128 Km.) The sub areas are, similarly to the 3x3 groups above, non-overlapping and congruent.

To determine which of the nadir zones the sub-areas fall into in the case of a sub-area not falling entirely into one of the zones defined by |X1| or |X2|, we use the parameters of the zone with the greatest overlap. For example, if a zone defined by Xa: Xb, where Xa < X1 < Xb

Use the zone -X1 to X1 if (X1-Xa) < (Xb-X1)

Use the zone |X1| to |X2| otherwise.

For each group do Steps 5.5.3.6.1-1 to 5.5.3.6.1-5 inclusive:

Step 5.5.3.6.1-1

Determine solar elevation (90-solar_zenith) at the centre of the across-track band which contains the central pixel of the group.

Transfer the 11µm data for the group being investigated into the 3 x 3 element array ir11

Step 5.5.3.6.1-2

Determine whether the pixels in group are over land or over sea, using the appropriate land flag (confidence word in global flags).

If one or more pixels in a group are over land then set the group_land_flag for the group and store for later use.

If there are 2 or fewer valid pixels, set the flag for the group in group_cloud_flag and proceed to the next group. A valid pixel has the 11 micron brightness temperature > 0 and <view>_fill_state = NATURAL_PIXEL as in step 5.5.3.6.1-3 below.

Step 5.5.3.6.1-3

Calculate the average and the standard deviation of the 11µm brightness temperatures for the valid pixels in the group, excluding cosmetic fill pixels, as follows. Filled pixels are identified by:

$$\langle \text{view} \rangle_fill_state(i, j) = NATURAL_PIXEL.$$

$$average_11 = (1/n) * \sum I(ir11, v; i, j)$$

$$sigma_11 = \sqrt{(1/(n-1)) * \sum (I(ir11, v; i, j) - average_11)^2}$$

where the sums are over the indices (i, j), the index of the pixels in the group that satisfy the selection criteria above, and n their total number.

Store the average_11 in average_11_array for later use.

Step 5.5.3.6.1-4

Select the appropriate threshold; if the group is over sea set threshold_sd to SEA_MAX_DEV. If the group is over land (i.e. if group_land_flag is set), inspect the solar elevation angle determined in Step 5.5.3.6.1-1. (Note that cells of mixed surface type are treated here as land cells.)

If solar_elevation > 5°

then set threshold_sd to the day-time value LAND_DAY_MAX_DEV; otherwise set threshold_sd to the night-time value LAND_NIGHT_MAX_DEV.

Step 5.5.3.6.1-5

If



$\text{sigma}_{11} > \text{threshold}_{sd}$
then set flag for the group in group_cloud_flag and proceed to the next group.

Step 5.5.3.6.1-6

When all the groups in a view have been processed investigate whether the spatial variation may result from temperature gradients on the surface rather than from cloud. For each group flagged as cloudy (i.e. when group_cloud_flag has been set) in step 5.5.3.6.1-5, do the following:

- i) If four or more of the up to 8 surrounding groups are clear (i.e. group_cloud_flag not set) and the group contains more than two valid pixels (as at step (b) above), then proceed to Step ii; otherwise move to the next flagged group.
- ii) If the central group is open sea (i.e. group_land_flag not set) and if it is not a border group, calculate the average difference between the 11 and 12 micron channel brightness temperatures for the central (cloudy) group ($\text{average}_{11_12_dif_cloudy}$) and for those of the surrounding groups that have not been flagged as cloudy ($\text{average}_{11_12_dif_clear}$)

Specifically,

$$\text{average}_{11_12_dif_cloudy} = \frac{1}{N_{cloud}} \sum_{i,j} (I(ir11, v; i, j) - I(ir12, v; i, j))$$

where the pixels indexed by i and j fall within the central cloudy group, are *NATURAL* pixels, and are valid in both channels: $I(ir11, v; i, j) > 0$ and $I(ir12, v; i, j) > 0$. N_{cloud} is the total number of pixels contributing to the sum. Similarly

$$\text{average}_{11_12_dif_clear} = \frac{1}{N_{clear}} \sum_{i,j} (I(ir11, v; i, j) - I(ir12, v; i, j))$$

where the pixels indexed by i and j fall within the clear groups, are *NATURAL* pixels, and are valid in both channels: $I(ir11, v; i, j) > 0$ and $I(ir12, v; i, j) > 0$. N_{clear} is the total number of pixels contributing to the sum.

If $\text{abs}(\text{average}_{11_12_dif_cloudy} - \text{average}_{11_12_dif_clear}) < \text{COHERENCE_RESET_THRESH}$,

then unflag the central group by setting to zero the appropriate element in group_cloud_flag .

Step 5.5.3.6.1-7

Flag all pixels in the cloud-flagged groups (i.e. where group_cloud_flag is set) for which the land flag is not set, as cloudy by setting the small scale cloud flag in each view.

[Note: the objective of the modification above is to effectively disable the spatial coherence test over land without introducing inadvertent side-effects. The cloud flags for each pixel are to be set to group_cloud_flag AND (NOT land). Of course for a given regridded pixel the nadir and oblique view land flags should be the same.]

Step 5.5.3.6.1-8

By using group_land_flag , loop through all the groups in view and flag groups within approximately 6 km of land, as well as all-land groups. A group's flag $\text{extended_land_flag}(x_index, y_index)$ is set if



$$\sum \sum \text{group_land_flag}(i, j) > 0$$

where the summations are for indices in the ranges of

$$x_index - 3 < u < x_index + 3 \text{ and}$$

$$y_index - 3 < v < y_index + 3$$

u and v must also be in the range of 0 to (L+1)., i.e. for groups that overlap the image boundary, the pixels out of range are not counted.

Step 5.5.3.6.1-9

Create arrays used in the large-scale coherence test.

- i) Determine the number of sub-arrays(sub_area_nx_v sub_area_ny) across the oblique and nadir views (v=NA or OB). The two views are treated differently in the x direction as the swath widths are different.

$$\text{sub_area_nx_v} = \text{width of swath(nadir or oblique view)} / \text{COH_AREA_SIZE_X}$$

and

$$\text{sub_area_ny} = \text{length of swath} / \text{COH_AREA_SIZE_Y}$$

- ii) Create arrays of size [sub_area_nx_v][sub_area_ny].

Step 5.5.3.6.1-10

For all sub-areas, find the group(s) with the highest average 11 μm brightness temperature, and the average (11 μm - 12 μm) brightness temperature difference(s) of these:

- i) Select sub-areas for all combinations of x_index_1_v, y_index_1, where v again denotes the view. The ranges of x_index_1_v and y_index_1 are 0 to (sub_area_nx_v - 1 or sub_area_ny - 1).

The groups for sub-area(x_index_1_v, y_index_1) are the array elements average(ix1_v:ix2_v, iy1:iy2) where

$$ix1_v = \text{INT}((x_index_1_v * \text{coh_area_size_x}) / 3)$$

$$ix2_v = \text{INT}(((x_index_1_v + 1) * \text{coh_area_size_x}) / 3) - 1$$

$$iy1 = \text{INT}((y_index_1 * \text{coh_area_size_y}) / 3)$$

$$iy2 = \text{INT}(((y_index_1 + 1) * \text{coh_area_size_y}) / 3) - 1$$

Step (5.5.3.6.1-10i) associates pixel groups with the sub-areas. The set of groups that corresponds to the sub-area indexed by x_index_1_v, y_index_1 comprises those groups whose indices x_index, y_index satisfy

$$(x_index, y_index) \in A(x_index_1_v, y_index_1), \text{ where}$$

the set of index pairs $A(x_index_1_v, y_index_1)$ is formally defined by

$$A(x_index_1_v, y_index_1) = \{(x_index, y_index): ix1_v \leq x_index \leq ix2_v; \\ iy1 \leq y_index \leq iy2\}$$

Since the number of groups is not a multiple of the sub_area_n, the sets $A(x_index_1_v, y_index_1)$ do not each contain the same number of groups.

For all sub-areas,

- ii) find the highest average 11 μm brightness temperature (max_bt_11) and its/their coordinates. Use valid groups only.

'valid' implies that the group satisfies the following conditions:

- its extended_land_flag is not set;
- it has passed the small-scale coherence test; that is, if group_cloud_flag is not set;
- it has not been shown to contain significant cloud by any previous test;



- it contains at least 3 natural pixels having valid 11 micron and 12 micron brightness temperatures; that is, at least 3 NATURAL pixels are available to contribute to the computation of the average (11 micron - 12 micron) brightness temperature difference at $j.3$ below.

Determine the flag `previous_tests(x_index, y_index)` as follows:

If `cloudy_box_thresh = -1`, the value of `previous_tests(x_index, y_index)` is determined by inspecting only the central pixel of the group:

```
previous_tests(x_index, y_index) =
  (<view>_cloud_state[i][j].gross_test = TRUE or
   <view>_cloud_state[i][j].thin_cirrus = TRUE or
   <view>_cloud_state[i][j].med_high_cloud = TRUE or
   <view>_cloud_state[i][j].fog_low_stratus = TRUE)
```

where i is the lesser of $(3*y_index + 1, 510)$ and j is the lesser of $(3*x_index + 1, 510)$; i and j index the central pixel of the group.

If `cloudy_box_thresh > 0`, the value of `previous_tests(x_index, y_index)` is determined by inspecting all pixels of the group:

```
previous_tests(x_index, y_index) = (n_cloudy ≥ cloudy_box_thresh)
```

where `n_cloudy` is the total number of pixels that fall within the group (x_index, y_index) and for which

```
(<view>_cloud_state[i][j].gross_test = TRUE or
 <view>_cloud_state[i][j].thin_cirrus = TRUE or
 <view>_cloud_state[i][j].med_high_cloud = TRUE or
 <view>_cloud_state[i][j].fog_low_stratus = TRUE).
```

The pixels that fall within the group defined by (x_index, y_index) are those whose indices satisfy

$$i_0 \leq i \leq i_0 + 2, \quad j_0 \leq j \leq j_0 + 2,$$

where i_0 is the smaller of $(3*y_index + 1, 509)$ and j_0 is the smaller of $(3*x_index + 1, 509)$.

Then

```
max_bt_11 = maximum value of average_11_array(x_index, y_index)
over the set of indices (x_index, y_index) ∈ A_valid(x_index_1, y_index_1)
```

The set $A_{valid}(x_index_1, y_index_1)$ is the set of indices of VALID groups, and is defined by

```
A_valid(x_index_1, y_index_1) = {x_index, y_index:
  (x_index, y_index) ∈ A(x_index_1, y_index_1)
  and NOT group_cloud_flag(x_index, y_index)
  and NOT extended_land(x_index, y_index)
  and NOT previous_test(x_index, y_index)
  and (valid_pixel_pairs ≥ 3)}
```

where `valid_pixel_pairs` is the number of pixels in the group defined by (x_index, y_index) for which `<view>_fill_state(i, j) = NATURAL_PIXEL`, and both $l(ir11, v; i, j) > 0$, $l(ir12, v; i, j) > 0$. The set of pixels in the group (x_index, y_index) is as defined above.

Then

```
sub_area_max_11(x_index_1, y_index_1) = max_bt_11.
```

It is possible that there are no valid groups; A_{valid} is an empty set. In this case `sub_area_max_11(x_index_1, y_index_1)` and `sub_area_dif(x_index_1, y_index_1)` should each be set to an exceptional value (≤ 0) to ensure that they are ignored in Step (m) below.



j.3) Calculate the average (11 μ m - 12 μ m) brightness temperature difference(s) for the groups with max_bt_11. Use valid pixels only, as in step c).

Use pixels with x coordinates in the range of
 $\max(0, 3*x_{\max} - 1)$ and $\min(511, 3*x_{\max} + 1)$
 and y coordinates in the range of
 $\max(0, 3*y_{\max} - 1)$ and $\min(511, 3*y_{\max} + 1)$

If a single group had the average 11 μ m brightness temperature value of max_bt_11 then assign its brightness temperature difference value to sub_area_dif

If more than one group has the same max_bt_11 value then find the highest of differences, and assign this value to sub_area_dif.

This step calculates the average difference between the 11 micron and 12 micron channels for those valid groups for which

$$\text{average_11_array}(x_index, y_index) = \text{max_bt_11}$$

Thus

sub_area_dif(x_index_1, y_index_1) = maximum value of
 (11 micron - 12 micron difference) over the groups having
 $(x_index, y_index) \in A_{\text{valid}}(x_index_1, y_index_1)$ and
 $\text{average_11_array}(x_index, y_index) = \text{max_bt_11}$

Step 5.5.3.6.1-11

k) Set the land_sub_area flag to 1 for all sub-areas in which one or more groups are over or near land i.e. set the flag if

$$\sum \text{extended_land_flag}(x_index, y_index) > 0$$

where the sum includes all $(x_index, y_index) \in A(x_index_1, y_index_1)$ as defined in step j.

Step 5.5.3.6.1-12

l) Set valid_sub_area_flag to 1 for all sub-areas for which the number of clear sea groups is greater than COH_FRACTION_PASSED of the total number, i.e.

$$\sum (1 - \text{group_cloud_flag}(x_index, y_index)) * (1 - \text{extended_land_flag}(x_index, y_index)) / (\text{COH_AREA_SIZE}/3)**2 > \text{COH_FRACTION_PASSED},$$

where the sum includes all $(x_index, y_index) \in A(x_index_1, y_index_1)$,

and sub_area_dif(x_index_1, y_index_1) is greater than COH_MIN_DIF.

Step 5.5.3.6.1-13

m) For invalid sub-areas, set the 11 μ m brightness temperature threshold threshold_11 to 32000 cK i.e. to an unrealistically high value so that all the sea pixels in these sub-areas are flagged as cloudy and omit Steps (m.1) to (m.7).

For sub-areas that are valid, determine threshold_11 as follows (Steps (m.1) to (m.7)):

m.1) Select the up to 9 sub-areas (valid or not) centred on the one being investigated. These are the sub-areas whose indices fall in the set

$$S(p, q) = \{(x_index_1, y_index_1): \quad 0 \leq x_index_1 < \text{sub_area_n}, \\ \quad 0 \leq y_index_1 < \text{sub_area_n},$$



$$p-1 \leq x_index_1 \leq p+1, q-1 \leq y_index_1 \leq q+1\}$$

where $p, q, 0 \leq p < sub_area_n, 0 \leq q < sub_area_n$, are the indices of the sub-area currently under consideration.

m.2) Set `land_in_areas` to 1 if one or more of the sub-areas selected has land:

$$land_in_areas = (\sum land_sub_area(x_index_1, y_index_1)) \geq 1$$

where the sum is over index pairs $(x_index_1, y_index_1) \in S(p, q)$.

Step (m.2) sets the flag `land_in_areas` if, for any $(x_index_1, y_index_1) \in S(p, q)$, `land_sub_area(x_index_1, y_index_1)` is set.

m.3) Find the highest `sub_area_dif`, `bt_dif_max`, for valid selected sub-areas.

$$bt_dif_max = \max (sub_area_dif(x_index_1, y_index_1)),$$

over the set of index pairs $(x_index_1, y_index_1) \in S(p, q)$ that identify valid sub-areas.

Notice that at least one sub-area (the central one) must be VALID, so this operation is always possible.

m.4) Select those `sub_area_max_11` values for which the corresponding `sub_area_dif` is within `difference_threshold` of `bt_dif_max`, found in step m.3. That is,

$$sub_area_dif(x_index_1, y_index_1) > bt_dif_max - difference_threshold.$$

The value of `difference_threshold` is given by:-

$$COH_AREA_DIF * (1 + land_in_areas * COH_ADJ_DIF_LAND)$$

m.5) Find the lowest `max_bt` of those sub-areas selected in step m.4

$$lowest_max_bt = \text{minimum of } sub_area_max_11(x_index_1, y_index_1)$$

over the set of index pairs such that $(x_index_1, y_index_1) \in S(p, q)$, the sub-area is valid, and `sub_area_dif(x_index_1, y_index_1)` satisfies the test in step m.4.

m.6) Calculate `threshold_11` for the sub-area investigated by decreasing `lowest_max_bt` by `COH_AREA_THRESH + land_in_areas * COH_ADJ_THRESH_LAND`.

That is:

$$threshold_11 = lowest_max_bt - COH_AREA_THRESH \\ - land_in_areas * COH_ADJ_THRESH_LAND$$

eq 5.5-61

m.7) If only one valid sub-area was selected at step m.4, and the `land_in_area` flag is set, lower `threshold_11` by 200 cK.

m.8) Flag all sea pixels in the sub-area investigated for those groups for which `average_11_array` is below `threshold_11` by setting the `spatial_coh` cloud flag in each view.

5.5.3.6.2 Gross cloud test

The test uses a look-up table with values of the threshold tabulated at intervals of 1° of latitude for each view (oblique and nadir), and for each calendar month. The test uses different look-up tables for the land and sea pixels.

Modifications to the AATSR test: The threshold depends on latitude, month and view. As above, we may need to divide the nadir swath into zones by across-track limits $X1, X2$ such that:

- if $-X1 < X < X1$ the AATSR nadir view constants are used;
- if $X1 < \text{abs}(X) < X2$, a new intermediate set of values are used;
- of $X2 < \text{abs}(X)$, the AATSR forward view constants are used.



- a) Derive the month in which the data was collected from the time of the ascending node.
- b) Do the gross cloud check for each valid pixel (over both land and sea) in each view ($v = n \mid f$) as follows:
 - b.1) Extract the pixel latitude $image_latitude(i, j)$ (in degrees), and convert to an integer $latitude_index$ in the range 0 to 180 by adding 90 and taking the integer part.
 - b.2) Determine threshold $gross_cloud_threshold_x_v$ (where x denotes land (lnd) or sea and v denotes na , int or ob for the view) using the latitude index just determined and the month to enter the gross cloud check threshold array for the relevant land/sea and view.
 - b.3) If the $12\mu\text{m}$ brightness temp $I(ir12, v; i, j) < gross_cloud_threshold$, flag the pixel as cloudy by setting the $12\mu\text{m}$ gross cloud test flag ($gross_test$) for each view.

5.5.3.6.3 Thin cirrus test

The test uses a look-up table that defines a threshold THIN_CIRRUS_THRESHOLD[temperature index][across-track band] for the nadir and oblique views separately, for each across-track band, and for values of the brightness temperature index, defined by $(T-250)$ K, from 0 to 60 inclusive. (Note that by symmetry the coefficients for the across-track band $i = 0, 1, \dots, 4$ are the same as for band $9 - i$.) The across track band dimensions are also included in the cloud LUT.

Modifications to the AATSR test: The index depends on across-track band, so we will need to define a wider set of across-track bands to cover the wider swaths.

It is applied to the oblique ($v = ob$) and nadir ($v = na$) views separately. For each pixel:

- a) Determine the pixel across track band number ($band_no$) using the across-track pixel number and the band table provided in the cloud LUT.

$$across_track\ band\ for\ pixel\ j = band_table[abs(j)]$$

- b) Determine the brightness temperature index using the $11\mu\text{m}$ value $I(ir11, v; i, j)$
 - $bt_index = \text{integer part of } (I(ir11, v; i, j) - 25000) / 100$
 - if ($bt_index > 60$) then $bt_index = 60$
 - else if ($bt_index < 0$) then $bt_index = 0$
 - end if

(Note that brightness temperature values are stored in units of 0.01 K)

- c) Calculate the difference
 - $I(ir11, v; i, j) - I(ir12, v; i, j)$

If this difference is greater than
THIN_CIRRUS_THRESHOLD_v[bt_index][band_no]

for the appropriate view v , flag the pixel as cloudy by setting thin_cirrus cloud.

5.5.3.6.4 Medium/high level cloud test

The test uses a look-up table that defines a threshold MED_HIGH_LEVEL_THRESH[temperature index] for the nadir and oblique views separately, for values of the brightness temperature index, defined by $2*(T-250)$ K, from 0 to 120 inclusive. The test is applied to the oblique and nadir views separately.



Modifications to the AATSR test: The threshold depends in brightness temperature and view. It therefore may be necessary to divide the nadir swath into zones by across-track limits X1, X2 such that:

if $-X1 < X < X1$ the AATSR nadir view constants are used;
if $X1 < \text{abs}(X) < X2$, a new intermediate set of values are used;
of $X2 < \text{abs}(X)$, the AATSR forward view constants are used.

- a) For each scan i , retrieve the solar elevation at each end of the scan. Calculate the solar elevation at each end of the scan using 90-solar zenith angle. The test is only performed if these are less than 5°: i.e. if
 - b) $\langle \text{view} \rangle_solar_elevation(i, 0) < 5.0$ or
 - c) $\langle \text{view} \rangle_solar_elevation(i, \text{max_j}) < 5.0$,
 - d) so excluding day-time data. Then for each pixel:
 - b) Determine threshold $\text{med_high_level_thresh}[\text{bt_index}]$ for the appropriate view from the table holding the threshold values for this test. The bt_index is calculated from

$$\text{bt_index} = \text{integer part of } (I(\text{ir}12, v; i, j) - 25000) / 50.$$
 if $(\text{bt_index} > 120)$ then $\text{bt_index} = 120$
 else if $(\text{bt_index} < 0)$ then $\text{bt_index} = 0$
 end if
 - (Note that brightness temperature values are stored in units of 0.01 K)
 - c) If the 3.7 and 12 micron brightness temperatures are valid, and the difference

$$I(\text{ir}37, v; i, j) - I(\text{ir}12, v; i, j)$$
 is greater than $\text{MED_HIGH_LEVEL_THRESH}[\text{bt_index}]$ for the appropriate view then flag the pixel as cloudy by setting the med_high_cloud flag in the respective view.

5.5.3.6.5 Fog/low stratus test

The test uses a look-up table that defines a threshold $\text{FOG_LOW_STRATUS_THRESHOLD}[\text{across-track band}]$ for the nadir and oblique views separately, for each across-track band as defined in the cloud LUT. (Note that by symmetry the coefficients for the across-track band $i = 0, 1, \dots, 4$ are the same as for band $9 - i$.)

Modifications to the AATSR test: The threshold depends on across-track band. A wider set of across-track bands will be necessary to cover the wider swath.

Apply the fog/low stratus test as follows:

- a) For each image scan i , retrieve the solar elevation at each end of the scan. Calculate the solar elevation at each end of the scan using 90-solar zenith angle. The test is only performed if these are less than 5°: i.e. if
 - $\langle \text{view} \rangle_solar_elevation(i, 0) < 5.0$ or
 - $\langle \text{view} \rangle_solar_elevation(i, \text{max } j) < 5.0$,
 - so excluding day-time data. Then for each pixel:
 - b) Determine the pixel across track band number (band_no) using the across-track pixel number and the band table provided in the cloud LUT.

$$\text{across-track band for pixel } j = \text{band_table}[\text{abs}(j)]$$



c) For each pixel for which a valid 11 micron and 3.7 micron brightness temperature is available, calculate the difference

$$I(ir11, v; i, j) - I(ir37, v; i, j).$$

If this difference is greater than FOG_LOW_STRATUS_THRESHOLD[band_no] for the appropriate view then flag the pixel as cloudy by setting the cloud flag fog_low_stratus.

5.5.3.6.6 Histogram test (1.6 micron)

The 1.6 μ m test is applied to sub-arrays of 32 \times 32 pixels. It uses the 1.6 μ m reflectance data from each of the nadir and oblique views.

Modifications to the AATSR test: This is a dynamic threshold test, so to the extent that the parameters do not depend on view, the only issue is the definition of the sub-array size. It is proposed to retain a size of 32 \times 32, so the higher resolution results in smaller cells. Provisionally the image size should be a multiple of 32 pixels. However, even if the swath width is not a multiple, the test should still work since the test allows for fewer pixels. This was the case for AATSR. If the number of pixels is minimum_for_histogram then the test currently returns a status of cloudy as the default.

The basis of this test is described by Zavody et al [RD-7]. The test relies on the fact that clouds have a higher reflectivity for short wavelength radiation than the sea (and, often, land). Following AATSR practice, the test is used only for measurements made over the sea, because land surfaces are, in general, much more reflecting than the sea, and show much higher variability in reflectivity. The separation of land and cloud by this test is hence less reliable. In addition, the processing scheme uses the test over coastal areas having both land and sea. Either these mixed areas would have to be completely rejected, or two separate tests performed for cloud contamination. The first option is not acceptable, and the second would require the development of sophisticated algorithms that are not merited.

The test uses a dynamic threshold to separate clear from cloudy pixels. Even over water surfaces, it is not possible to parameterize the 1.6- μ m signal from a clear sea area accurately by the viewing and incidence angles alone, since, in non-sun-glint conditions, a major source is scattering by atmospheric aerosols, the concentration of which depends on location, wind direction, etc. The threshold value of the signal for clouds is therefore determined dynamically by using the histogram technique as developed by Saunders, 1986.

The basis of the technique is that the 1.6 micron signal is much lower for clear sea areas than for cloudy pixels; moreover for clear sea and for areas of up to about 1000 km², the effect of atmospheric variability on the 1.6 μ m signal is usually small, while the 1.6 μ m signal for cloudy pixels is usually high and highly variable. It follows that if a histogram is formed using the 1.6 μ m signal, the clear sea pixels will give rise to a narrow peak at the low end of the histogram, with cloudy pixels distributed over many histogram bins above. The value of the 1.6 μ m signal at the peak, if it satisfies the criteria associated with clear pixels, can be used to derive the threshold to be used for the area.

If the area under study is close to the point of specular reflection from the sun - 'sun glint' - then the reflected radiation is high and the area cannot be distinguished from cloud on the relative strength of the signal alone. It is first of all necessary to identify these areas, and this is done by calculating the angle through which the surface would need to be tilted for full specular reflection to occur. If this tilt angle is below a specified threshold then a spatial coherence test, similar to that using the 11 μ m brightness temperature values, is used instead of the histogram technique.



Areas adjacent to those where glint occurs are treated specially by the histogram test. The 1.6 μm signal may have a significant gradient with respect to across-track distance even over the relatively small areas used, hence, if such near-glint conditions occur, the spatial gradient in the 1.6 μm values is removed by de-trending the data before the cloud threshold is determined and used on the reflectance data.

Elimination of sun-glint

The test cannot be applied if the pixels are affected by sun-glint, and so it is necessary to identify affected pixels. The test for sun-glint makes use of the solar and viewing angles computed by the geolocation process. Sun-glint will occur in the vicinity of the point at which specular reflection by a horizontal surface would direct incident solar radiation towards the instrument. The criterion for this to occur is that the vector that bisects the angle between the line of sight to the sun from the pixel, and that to the satellite from the pixel is vertical. This vector is in the direction of the normal to a surface facet that is orientated so as to reflect sunlight towards the instrument; thus if it is close to the vertical, surface facets may be correctly oriented to give specular reflection.

Consider a pixel at latitude φ , longitude λ , and let the solar azimuth and elevation at this point be α_{sun} , ε_{sun} , and the azimuth and elevation of the satellite as seen from the pixel be α_{sat} , ε_{sat} . Then the vectors

$$\mathbf{v}_{sun} = (\cos \alpha_{sun} \cos \varepsilon_{sun} \quad \sin \alpha_{sun} \cos \varepsilon_{sun} \quad \sin \varepsilon_{sun}) \quad \text{eq 5.5-62}$$

and

$$\mathbf{v}_{sat} = (\cos \alpha_{sat} \cos \varepsilon_{sat} \quad \sin \alpha_{sat} \cos \varepsilon_{sat} \quad \sin \varepsilon_{sat}) \quad \text{eq 5.5-63}$$

represent the lines of site from the pixel to the sun and to the satellite respectively. A surface facet whose normal bisects these lines of sight would be correctly orientated for specular reflection of sunlight towards the instrument. The vector $\mathbf{v}_{sun} + \mathbf{v}_{sat}$ bisects these two directions, and so if \mathbf{k} is the unit vector in the direction of the local vertical, the tilt angle required for specular reflection is

$$\tau = \arccos \left(\frac{\mathbf{k} \cdot (\mathbf{v}_{sun} + \mathbf{v}_{sat})}{|\mathbf{v}_{sun} + \mathbf{v}_{sat}|} \right) \quad \text{eq 5.5-64}$$

The unit vector in the vertical direction $\mathbf{k} = (0 \quad 0 \quad 1)$.

An area is considered to be affected by sunglint if its tilt angle calculated as above is less than 15 degrees. The range of tilt angles between 15 and 23 degrees constitutes the near-glint region, and areas within this range are treated specially by the algorithm.

Overview of algorithm

The auxiliary parameters used in this test are listed in Table 5-8.. Table 5-8 also gives the standard values of these parameters used for AATSR processing, although different values may be appropriate for SLSTR. Apart from the *PEAK_FACTOR* and *SPREAD* parameters, the same parameters are used for both oblique and nadir views.

Table 5-8: Auxiliary parameters used by the 1.6 micron tests

Parameter	Value
<i>SPREAD_NA</i>	0.4
<i>SPREAD_OB</i>	0.6
<i>PEAK_FACTOR_NA</i>	4
<i>PEAK_FACTOR_OB</i>	3



<i>MIN_FOR_HISTOGRAM</i>	64
<i>TILT_THRESHOLD</i>	15
<i>THRESHOLD_3</i>	0.07
<i>NEAR_GLINT_RANGE</i>	8.0
<i>MIN_FOR_PASSED</i>	64
<i>SEARCH_RANGE_FOR_PEAK</i>	1.5
<i>TILT_WEIGHT_LIMIT</i>	23
<i>RANGE_WEIGHT_LIMIT</i>	40
<i>TILT_WEIGHT_FACTOR</i>	0.025
<i>MIN_PEAK_VALUE</i>	10
<i>MIN_FOR_DETREND</i>	256
<i>MAX_GLINT_THRESHOLD</i>	90.0

The 1.6 µm test is applied to the nadir and oblique images independently. It is assumed to operate on image segments of 512 image rows, and it is only applied to daytime scenes for which the solar elevation at the centre of the image segment exceeds 5 degrees. For each view the image segment is divided into sub-arrays of 32 by 32 pixels, each representing an area approximately 32 km square and containing a maximum of 1024 valid pixels, and the test is applied to each sub-array in turn.

The sub-area is tested for sunglint. The solar and satellite azimuth and elevation at the centre of the 32 by 32 pixel sub-area are calculated by linear interpolation between the band edge values, and the tilt angle is calculated using (5.5-127) to (5.5-129). If the tilt angle is less than the auxiliary parameter *TILT_THRESHOLD*, sunglint may be present, and the histogram test is not applied. Instead a spatial coherence test is applied as described in a later section. If the tilt angle is less than (*TILT_THRESHOLD* + *NEAR_GLINT_RANGE*), the sub-area falls in the near-glint region.

If the sub-area is not affected by sunglint, the histogram of the 1.6 micron reflectance values is generated for the set of valid clear sea pixels in the sub-array. To be included in the histogram a pixel must satisfy all three of the following criteria:

- the 1.6µm reflectivity should be non-negative;
- it should not be a land pixel;
- it should not previously have been flagged as cloudy by the gross cloud test, the infra-red spatial coherence test or the thin cirrus test. (These daytime tests precede the 1.6 micron test.)

The histogram is generated with a bin size of 0.1%. If the number of valid pixels contributing to the histogram is less than *MIN_FOR_HISTOGRAM*, these pixels are all flagged as cloudy and processing moves on to the next sub-array. Otherwise, the lower end of the histogram is searched for a valid histogram peak.

The peak is defined by the maximum value, *peak_value*, in the range of bins between *low_interval* and *low_interval* + 15, where *low_interval* is the index of the first non-zero histogram bin. The position of the histogram peak, *peak_interval* is calculated to sub-bin accuracy by fitting a quadratic function to the maximum value and the two adjacent ordinates and finding the position of the maximum of the quadratic; the algebra here is the same as that described below for the infrared histogram test.

If the current sub-array lies outside the near-glint region, the peak is accepted as a valid clear sea peak if the following three conditions are satisfied:

$$peak_interval - low_interval < SPREAD \quad \text{eq 5.55.5-65}$$

$$peak_value > MIN_PEAK_VALUE \quad \text{eq 5.5-66}$$

$$peak_value > (PEAK_FACTOR + f) \times average_count \quad \text{eq 5.55.5-67}$$



In equation (5.5-132) *average_count* represents the average value of the histogram over the range for which it is non-zero, and *f* is an adjustment factor depending upon the histogram range. The basis of the latter is described by Zavody et al (2000). The auxiliary parameter *SPREAD* is that (*SPREAD_NA* or *SPREAD_OB*) appropriate to the view.

If the conditions are satisfied, a reflectance threshold is set at the value

$$reflectance_threshold = peak_interval + 0.5 \times SPREAD. \quad eq\ 5.55.5-68$$

All valid pixels in the sub-area having a reflectance value greater than *reflectance_threshold* are flagged as cloudy.

If the conditions are not satisfied, the peak may be cloud-contaminated. In this case all the valid formerly clear sea pixels are flagged as cloudy and processing continues to the next sub-array.

The above is the basic algorithm applied outside near-glint regions. If the current sub-array lies within the near-glint region, the processing is modified. Firstly, the procedure above for determining the threshold is modified, and secondly, the pixel reflectance values are de-trended to remove residual glint reflectance before the threshold is applied.

For near-glint regions, the quantities *PEAK_FACTOR* and *SPREAD* in (5.5-130) and (5.5-132) are modified.

A new quantity

$$SPREAD_ADJUSTED = SPREAD \times (1 + (1 - \tau) TILT_WEIGHT_LIMIT)) \quad eq\ 5.55.5-69$$

is derived, and the function

$$g(\tau) = 1 - (TILT_WEIGHT_LIMIT - \tau) \times TILT_WEIGHT_FACTOR \quad eq\ 5.55.5-70$$

is calculated. Equations (5.5-30) and (5.5-132) are replaced by

$$peak_interval - low_interval < SPREAD_ADJUSTED \quad eq\ 5.5-71$$

$$peak_value > (PEAK_FACTOR + f) \times g \times average_count \quad eq\ 5.55.5-72$$

If the three conditions (5.5-136), (5.5-131), (5.5-137) are satisfied, a reflectance threshold is then set as

$$reflectance_threshold = peak_interval + 0.5 \times SPREAD_ADJUSTED. \quad eq\ 5.55.5-73$$

The residual solar glint may lead to erroneous flagging of clear pixels towards the east side of the swath. To compensate for this effect a new histogram is generated after de-trending the data. The across-track reflectance gradient is calculated by fitting a regression line to the plot of reflectance against across-track distance. The data is corrected by subtracting a line of this slope from the reflectance values, and a new histogram is generated from the resulting detrended data. This is tested as before, and a new reflectance threshold derived. If the new histogram peak is valid, all valid pixels in the sub-area having a detrended reflectance value greater than *reflectance_threshold* are flagged as cloudy. Otherwise a spatial coherence test is attempted.

5.5.3.6.6.1 Spatial coherence test

The sub-array is divided into rectangular groups of 2 pixels in the across-track dimension by 4 pixels oblique. The short dimension of the group aligned in the across-track direction to minimise the effects of across-track gradients of reflectance. The spatial coherence test compares the standard deviation of the reflectance on pixels in the group (omitting cosmetically filled pixels) with a threshold, and flagging those as cloudy that exceed the threshold. The threshold is derived by finding the pixel in the sub-array that has



the highest brightness temperature in the 12 micron channel and finding the standard deviation of the 3 by 3 pixel group centred on this.

5.5.3.6.6.2 Detailed Structure of the Algorithm

The test is performed for the nadir and forward view separately. We define L as the size of the sub-array for which the histogram is derived. For each view the following steps are carried out for each non-overlapping L by L pixel sub-array within the 1.6 micron reflectance image. (The histogram will thus contain a maximum of $L \times L$ pixels.)

1) Verify that the sub-array includes sufficient valid pixels to perform the test. Find the total number of valid pixels; if this is less than $MIN_FOR_HISTOGRAM$ then flag all pixels as cloudy, and continue to the next sub-array. In this algorithm 'valid' implies that a pixel satisfies the following conditions:

- it has a valid 1.6 μ m reflectance value (greater than zero);
- it is not over land;
- it has not been flagged by the gross test or the spatial coherence test or the thin cirrus test.

2) Check for sun glint:

2.1) Use linear interpolation with respect to across-track distance to derive the values of the following solar and viewing angles at the mid-point of the subgroup, using the solar and viewing angles appropriate for the view (as calculated in Section 0): solar elevation angle; satellite elevation angle; satellite azimuth; solar azimuth.

2.2) Calculate the tilt angle τ , the angle by which the sea surface would need to be tilted to give specular reflection, using equations (5.5-153) to (5.5-155) above.

2.3) If the tilt angle $\tau < TILT_THRESHOLD$ then sun glint is deemed to be present. If this condition is satisfied then set a *glint_present* flag.

2.3.1) If *glint_present* is not set then generate a histogram using the 1.6- μ m reflectance values with a bin size of 0.1% (i.e. 10 product units), and continue at Step 3.

2.3.2) If *glint_present* is set then set the sunglint flag for the pixels in the sub-array and go to Step 7.

3) Determine the following properties of the histogram:

low_interval the reflectance of the first bin with at least one pixel contributing;
high_interval the reflectance of the last bin with at least one pixel contributing;
hist_range ($high_interval - low_interval$) + 0.1;

(Note that the units of these quantities are % reflectance; this is related to the histogram bin index by dividing the latter by 10.0.)

peak_box_no the box number at the histogram peak, within the reflectance range of *low_interval* and ($low_interval + SEARCH_RANGE_FOR_PEAK$).

peak_value the histogram value at bin *peak_box_no*

average_value the average number of pixels per bin for the histogram, between the reflectance limits of *low_interval* and *high_interval*

By using the three histogram values centred on *peak_box_no*, calculate the accurate peak *peak_interval* position. Fit a quadratic function at the 3 points, and find the abscissa of the function maximum. This, expressed in units of percent, is *peak_interval*.

4) Set the parameter *SPREAD* to *SPREAD_NA* or *SPREAD_OB* according to whether the nadir or the oblique view is selected. Similarly set *PEAK_FACTOR* to *PEAK_FACTOR_NA* or *PEAK_FACTOR_OB* as appropriate. Note: because of the wider nadir swath of SLSTR compared to AATSR it may be



necessary that *SPREAD_NA* and/or *PEAK_FACTOR_NA* must be specified as functions of across-track position. This is to be reviewed. This complication does not arise for the oblique view.

Calculate the following:

If *hist_range* < *RANGE_WEIGHT_LIMIT* then

$$f = (1 - \text{PEAK_FACTOR}) \times \text{sqrt}((\text{RANGE_WEIGHT_LIMIT} - \text{hist_range}) / (\text{RANGE_WEIGHT_LIMIT} - 0.2))$$

otherwise set $f = 0$.

If $\tau < \text{TILT_WEIGHT_LIMIT}$ then $g = 1 - (\text{TILT_WEIGHT_LIMIT} - \tau) \times \text{TILT_WEIGHT_FACTOR}$, otherwise $g = 1.0$

If $\text{tilt} \geq \text{TILT_WEIGHT_LIMIT}$, then set *SPREAD_ADJUSTED* = *SPREAD*.

If $\text{tilt} < \text{TILT_WEIGHT_LIMIT}$, then set

$$\text{SPREAD_ADJUSTED} = \text{SPREAD} \times (1 + (\text{TILT_WEIGHT_LIMIT} - \tau) / \text{TILT_WEIGHT_LIMIT})$$

Test whether or not the histogram satisfies the following conditions:

$$(\text{peak_interval} - \text{low_interval}) < \text{spread_adjusted}$$

$$\text{peak_value} > (\text{peak_factor} + f(\text{hist_range})) \times g(\tau) \times \text{average_value}$$

$$\text{peak_value} > \text{MIN_PEAK_VALUE}$$

4.1) If all three conditions are satisfied then set the reflectance threshold:

$$\text{reflectance_threshold} = \text{peak_interval} + \text{SPREAD_ADJUSTED}/2.$$

Test whether

$$\text{TILT_THRESHOLD} < \tau < (\text{TILT_THRESHOLD} + \text{NEAR_GLINT_RANGE})$$

4.1.1) If yes, then the area is in a near-glint region; go to Step 5.

4.1.2) If no, go to Step 6

4.2) If any of the three conditions is not satisfied then test whether the area is in a near-glint region.

4.2.1) If the area is not in a near-glint region, as defined in 4.1 above, then flag all pixels as cloudy and continue to the next sub-array of 32 x 32 pixels.

4.2.2) If the area is in a near-glint region then do calculate

$$n = 1.5 \times \text{SPREAD_ADJUSTED}$$

Calculate the total number of valid pixels with reflectance values between *low_interval* and *low_interval* + *n* and test whether this number exceeds *MIN_FOR_DETREND*.

4.2.2.1) If yes, set *reflectance_threshold* = *low_interval* + *n* and go to Step 5.

4.2.2.2) If no, go to Step 7 (spatial coherence test on reflectance values).

5) De-trend the reflectance data before applying a threshold as follows:

5.1) Calculate the across-track gradient of reflectance, *gradient_16* by regressing the reflectance values against their across-track distance, using only pixels having reflectance values less than *reflectance_threshold*.

The gradient is obtained by computing, using valid pixels only,

$$a = \text{total_valid} \times \Sigma(x(i) \times ir16(i)) - \Sigma x(i) \times \Sigma ir16(i)$$

$$b = \text{total_valid} \times \Sigma x(i)^2 - (\Sigma x(i))^2$$

where *ir16* is the 1.6 micron reflectance of the pixel *i* and *x(i)* is its across-track co-ordinate. The summations are for total_valid values for the valid pixels only, and *i* takes the indices of valid pixels determined in step 1.

If *b* is not equal to zero then *gradient_16* = *a/b*, otherwise *gradient_16* = 0.

5.2) Correct the reflectance value of each valid pixel using the gradient calculated above:



$$detrended_16(i) = ir16(i) - gradient_16 \times x(i)$$

5.3) Form a new histogram using the detrended reflectances as in step 2.3.1 above.

5.4) Determine *low_interval* and *peak_interval* from the new histogram as in step 3

5.5) If $(peak_interval - low_interval) \geq SPREAD$ then go to step 7.

5.6) Determine the reflectance threshold appropriate to the de-trended data:
 $reflectance_threshold = peak_interval + SPREAD/2$.

6) Flag all the pixels with (de-trended, if appropriate) reflectance values greater than *reflectance_threshold* as cloudy, then continue at step 9.

5.5.3.6.6.3 Spatial Coherence Test

7) Calculate the standard deviation threshold for the 1.6 micron spatial coherence test *sd_threshold* in the following way:

7.1) Find the indices of the highest 12- μ m brightness temperature among the pixels in the 32×32 array, using only pixels that satisfy the following conditions:

- The pixel satisfies the validity conditions in Step 1;
- its reflectance at 1.6 μ m is less than *MAX_GLINT_THRESHOLD*
- it has not been cosmetically filled.

7.2) Set *reflectance_at_12um_max* equal to the 1.6 μ m reflectance value of this pixel (or, if more than one pixel is found, to that with the highest 1.6 μ m value).

7.3) The standard deviation threshold, in product units of 0.01%, is given by
 $sd_threshold = 100 \times reflectance_at_12um_max \times THRESHOLD_3$

8) Loop through the subgroup by examining the standard deviations of valid pixels from non-overlapping groups of (2 x 4) pixels and flag those whose standard deviation exceeds *sd_threshold*.

8.1) Calculate the standard deviation of 1.6 micron reflectance for the sixteen (2 by 4) pixel groups in horizontal strips of 4 pixels by 32 pixels. (The long side of the 2 by 4 rectangle is in the oblique direction.). Only use valid pixels as re-defined at 7.1 above.

8.1.1) Mark as cloudy those (2 x 4) groups having less than two valid pixels.

8.1.2) Mark as cloudy those (2 x 4) groups whose standard deviation exceeds *sd_threshold*.

8.1.3) If an unmarked (i.e. clear) group lies between two groups marked as cloudy in the strip then mark the group as cloudy.

Repeat these steps for all 8 strips in the 32 by 32 sub-array

8.2) Flag as cloudy all those pixels that are in a group marked as cloudy, and also any pixels with reflectance values at 1.6 microns in excess of *MAX_GLINT_THRESHOLD*.

9) Count the number of clear pixels remaining in the (32 x 32) array. If the total is less than *MIN_FOR_PASSED* then flag these also as cloudy.

5.5.3.6.7 11/12 micron nadir/oblique test

This test is only applied to sea pixels. It uses the auxiliary parameters *a0*, *a1* and *11_12_view_diff_thresh* given in the 11/12 micron LUT.

Modifications to the AATSR test: This test is applied only within the across-track swath. The test parameters depend on across-track band, so we must define a wider set of across-track bands to cover the wider swath.

Loop through the pixels in the *n_band* across-bands of an image (defined in the cloud LUT) and check the relationship between the measured brightness temperature differences for each pixel, using the appropriate parameters and threshold. Use the nadir view 12 micron and 11 micron brightness



temperatures and the oblique view 11 μ brightness temperatures. For each pixel i, j perform steps (a) to (d) inclusive if

the nadir land flag (i, j) (provided in the confidence parameter in global flags) is FALSE and if the brightness temperatures used in Steps (a) and (b) are valid:

a) Derive the nadir view (11 μ m - 12 μ m) difference

$$dif_11_12 = I(ir11, n; i, j) - I(ir12, n; i, j)$$

b) Derive the 11 μ m (nadir-view - oblique-view) differences

$$dif_nv_fv = I(ir11, n; i, j) - I(ir11, f; i, j)$$

c) By using the appropriate coefficients for the band of every pixel, given by equation 5.6.1, compute the expected 11 μ m (nadir-view - oblique-view) differences, given by

$$exp_dif_nv_fv = a0 + a1 * dif_11_12$$

where

$$a0 = a0(band_no);$$

$$a1 = a0(band_no)$$

d) Flag those pixels as cloudy for which the difference

$$abs(exp_dif_nv_fv - dif_nv_fv) > 11_12_view_diff_thresh$$

by setting the 11_12_view_diff cloud flag.

5.5.3.6.8 11/3.7 micron nadir/oblique test

If night-time and the 3.7 μ m channel is available, repeat the preceding test, now using the brightness temperatures measured in the 3.7 μ m and 11 μ m channels with the appropriate parameters and threshold. Use the nadir view 11 μ m and 3.7 μ m brightness, and the oblique view 3.7 μ m brightness temperatures. The test is only applied to sea pixels. This test uses the auxiliary parameters $b0, b1, b2$ and 37_11_view_diff_thresh provided in the 11/3.7 cloud test LUT.

Modifications to the AATSR test: This test is applied only within the across-track swath. The test parameters depend on across-track band, so we must define a wider set of across-track bands to cover the wider swath.

The test is only applied to pixels on scans for which

$$nadir_solar_elevation(i, 0) < 5.0 \text{ and}$$

$$nadir_solar_elevation(i, \max j) < 5.0 \text{ and}$$

$$along_track_solar_elevation(i, 0) < 5.0 \text{ and}$$

$$along_track_solar_elevation(i, \max j) < 5.0.$$

For each scan i for which the above condition is true, and for each pixel j , perform steps (a) to (d) inclusive if

the nadir land flag(i, j) is FALSE and if

the brightness temperatures used in Steps (a) and (b) are valid:

a) Derive the nadir view (3.7 μ m - 11 μ m) difference

b) $dif_37_11 = I(ir37, n; i, j) - I(ir11, n; i, j)$

b) Derive the 3.7 μ m (nadir-view - oblique-view) difference

c) $dif_nv_fv = I(ir37, n; i, j) - I(ir37, f; i, j)$

c) By using the appropriate coefficients for the band of every pixel, given in the cloud LUT data product, compute the expected 3.7 μ m (nadir-view - oblique-view) differences, given by

$$exp_dif_nv_fv = b0 + b1 * dif_37_11 + b2 * (dif_37_11)**2$$

where

$$b0 = b0(band_no);$$

$$b1 = b1(band_no);$$

$$b2 = b3(band_no).$$

d) Flag those pixels as cloudy for which the difference

$$abs(exp_dif_nv_fv - dif_nv_fv) > 37_11_view_diff_thresh$$

by setting the 37_11_view_diff cloud flag in the appropriate view.



5.5.3.6.9 Infrared histogram test

The histogram test operates on a 512 by 512 pixel image segment, so the orbit should be divided into image segments of this size; it should be imagined as 'tiled' with 512 row image segments. Note that if at the end of an orbit or for any other reason the data is insufficient to form a complete image segment of 512 rows the histogram can be based on an incomplete image segment. (Because histograms are made up of those pixels which are valid sea pixels, if an image is largely over land, one has an incomplete image anyway.) Provided the number of pixels exceeds MIN_FOR_11_12_HISTOGRAM the test can be applied. It is expected that processing will always include an integer number of granules, so if there is an incomplete image segment at end of data, it will always have at least 32 rows.

Note this test only applies to pixels over sea.

Modifications to the AATSR test: The size of the image segments to tile the views must be modified to HISTOGRAM_AREA_WIDTH by HISTOGRAM_AREA_LENGTH, chosen to fit the wider swaths. For the oblique view, HISTOGRAM_AREA_WIDTH will simply be the entire swath width. The nadir view swath is much wider and should be divided into three regions. A parameter X1 is defined in the cloud LUT and so when $-X1 < X < X1$, the nadir view parameters should be used. For $|X| > X1$, the forward view parameters should be used. HISTOGRAM_AREA_LENGTH can be set as 512 for both views.

This is the last of the cloud tests performed and it is applied only to those pixels that have passed all the previous cloud tests. It uses the 11 μm and 12 μm channels, and relies mostly on the following:

1. The difference between brightness temperatures measured in different channels is almost entirely determined by the atmospheric characteristics.
2. The atmospheric variability is usually small over the area of an AATSR image product. It follows that even when strong temperature gradients are present in the sea, the brightness temperature differences form a fairly tight cluster for the clear pixels.
3. Low stratus clouds, the most likely contamination to have escaped detection by the other tests, affect the brightness temperatures differently from the clear atmosphere. There are two effects:
 - a. The infrared radiation detected originates from a height above sea level and hence the lowest layer of the troposphere is only partially traversed by the radiation. As it is this layer that contains most of the atmospheric water vapour, which is the major source of the difference between the 11 μm and 12 μm brightness temperatures, these differences are reduced over clouds.
 - b. Low clouds have a slightly higher emissivity in the 12 μm channel than in the 11 μm channel. It follows that the proportion of reflected cold radiation is less for the 12 μm channel as for the 11 μm channel and hence the 12 μm brightness temperature over this type of cloud, compared with the 11 μm brightness temperature, is not as cold as over the clear sea.

The 12 μm brightness temperature is almost always lower than the 11 μm brightness temperature owing to the water vapour absorption. Both the above effects reduce the difference found over clear sea.

4. If the water vapour column amounts are not uniform over an image, and the sea surface temperature is constant, then the lowest brightness temperatures originate from pixels having the highest water vapour loadings and hence the highest (11 μm minus 12 μm) brightness temperature differences. It follows that, in this case, the brightness temperatures and the brightness temperature differences are negatively correlated for clear pixels and positively correlated for cloudy pixels. (N.B. Although this is generally true there are exceptions. A positive



correlation in SST with water vapour loading, or lower tropospheric air warmer than the SST, can reduce the correlation or even change its sign.)

The auxiliary parameters used in this test are listed in Table 5-7. They are taken from the auxiliary file of Cloud Look-up Table Data.

Table 5-7: Auxiliary parameters used by the infrared histogram test

<i>MIN_FOR_11_12_HISTOGRAM</i>	100.00
<i>PEAK_FRAC_MIN</i>	0.51
<i>LATITUDE_THRESH</i>	40.0
<i>SECOND_LOW_FRACTION</i>	0.50
<i>HALF_WIDTH_M_NA</i>	0.5
<i>HALF_WIDTH_B_NA</i>	25.0
<i>HALF_WIDTH_M_OB</i>	1.0
<i>HALF_WIDTH_B_OB</i>	15.0
<i>MAX_DIF_AVE_CHAN_1</i>	100.0
<i>MAX_DIF_PEAK_CHAN_1</i>	50.0
<i>RATIO_B</i>	4.0
<i>IR_SPREAD_NA</i>	60.0
<i>IR_SPREAD_OB</i>	80.0
<i>SLOPE_MAX_ALLOWED</i>	10.0
<i>IR_PEAK_MIN</i>	25000.0

5.5.3.6.9.1 Determination of the Difference Threshold

Thus the overall objective of the algorithm is simple, to determine a dynamic threshold against which the brightness temperature differences can be compared to differentiate cloudy from clear pixels. The determination of this threshold is based on the histogram of the distribution of the differences in brightness temperature between the 11 and 12 micron channels. In the presence of undetected cloud, the histogram is expected to be the superposition of two peaks, representing the distribution of the brightness temperature differences of clear sea and of cloudy pixels respectively.

The algorithm attempts to identify the presence of the two peaks, and to identify which one represents clear sea. In the simplest terms, the clear sea pixels should show a narrower and more regular distribution, at a higher brightness temperature difference, than the cloudy pixels. However, the peaks cannot be distinguished by their relative height, since the height of each peak depends on the number of contributing pixels, which is arbitrary and simply reflects the area of clear sea or undetected cloud as appropriate. Moreover, the peaks may overlap.

Therefore the peaks are tested against various criteria to identify a valid peak, that is, a peak that shows the characteristics of the distribution of clear sea pixels. If a valid peak is identified, a threshold is set on its lower edge, and all pixels having a brightness temperature difference less than this threshold are flagged as cloudy.

The procedure involves the following stages.

- 1 A histogram representing the frequency distribution of the brightness temperature difference ($T_{11} - T_{12}$) is prepared over all the sea pixels that have not been flagged as cloudy by any previous test. At the same time, for each histogram bin, the cumulative brightness temperature in the 12 micron channel is calculated over all the pixels that contribute to each bin.

The histogram is generated of the brightness temperature difference $T_{diff} = (T_{11} - T_{12})$ in the range $-20.0 \text{ K} \leq T_{diff} < 80.0 \text{ K}$ with a bin size $dT = 0.1 \text{ K}$. If the histogram bins are indexed by k ($0 \leq k < 1000$) then the



lower limit of bin k corresponds to $T = k \times dT - 20.0$ K. The corresponding histogram value $h(k)$ represents the number of clear sea pixels for which

$$k \times dT - 20.0 \leq (T_{11} - T_{12}) < (k + 1) \times dT - 20.0$$

The cumulative brightness temperature for bin k is ΣT_{12} , where the sum is over all contributing pixels that fall in bin k .

2 The maximum value of the histogram is found, and its position is taken to represent the principal peak of the histogram. The extent of this peak is determined, and the parameters, height, width and position that describe the peak are calculated.

The maximum of the histogram defines the principal peak, referred to in the following as the major peak, and corresponds to the mode of the frequency distribution. If the bin index corresponding to the maximum is $peak_interval[MAJOR]$, then the maximum value is

$$peak_value[MAJOR] = h(peak_interval[MAJOR]).$$

The extent of the peak is determined by finding the first local minimum or zero value, whichever is found first, on either side of the peak. The indices of these bins are $lower_limit[MAJOR]$ and $higher_limit[MAJOR]$.

3 The peak is tested against certain criteria to determine whether or not it is valid, that is, possesses the characteristics likely to represent clear pixels.

The peak is flagged as invalid if it corresponds to a difference value less than zero, that is, $peak_interval[MAJOR] < 200$, unless the absolute latitude at the centre of the image segment exceeds a threshold $LATITUDE_THRESH$ (currently set at 40 degrees) or the number of pixels $peak_value[MAJOR]$ exceeds a threshold IR_PEAK_MIN (currently 25000). Otherwise it is flagged as valid.

If the histogram represents a night-time scene and if the peak has passed the previous tests, so is flagged as valid, a further test is applied to verify that the histogram is not the sum of a clear and a cloudy histogram and hence excessively broad.

Firstly, a more accurate value of the histogram peak is determined, by fitting a quadratic curve to the peak value and the adjacent histogram values on either side. If the histogram peak and its neighbours are

$$\begin{aligned} h_0 &= h(peak_interval[MAJOR] - 1) \\ h_1 &= h(peak_interval[MAJOR]) \\ h_2 &= h(peak_interval[MAJOR] + 1), \end{aligned}$$

then the quadratic function is

$$\begin{aligned} h(x) &= 0.5x(x-1)h_0 - (x-1)(x+1)h_1 + 0.5x(x+1)h_2 \\ &= 0.5x^2(h_0 - 2h_1 + h_2) + 0.5x(h_2 - h_0) + h_1, \end{aligned}$$

where x is a normalised abscissa co-ordinate measured from $peak_interval$. The maximum of this function occurs at

$$x = 0.5(h_0 - h_2)/(h_0 - 2h_1 + h_2).$$

Thus provided the denominator of this expression is not zero, which can only occur if all three ordinates are the same, improved values of the histogram peak and its position are found. The improved peak value $exact_peak_value$ is obtained by evaluating equation () at the value of x given by equation (), while the new position is



$$\begin{aligned} \text{exact_peak_interval} &= (\text{peak_interval} + x) \\ \text{ir11_ir12_diff_at_peak} &= (\text{exact_peak_interval} - 200) * dT \end{aligned}$$

Next, the width of the histogram at half-height is computed to sub-bin accuracy by interpolation between the two histogram values bracketing the 50% peak values defined by $0.5 * \text{exact_peak_value}$ and differencing the abscissae. A half-width threshold is computed from

$$\text{half_width_threshold} = M \times \text{ir11_ir12_diff_at_peak} + B,$$

where M and B are constants defined in the auxiliary file of cloud LUT data, `ATS_CL1_AX`. (These constants have different values for the oblique and nadir views.)

The peak is flagged as invalid if

$$\text{half_width}[\text{MAJOR}] > \text{half_width_threshold}.$$

4 A new histogram is determined by removing the peak just found (the major peak) by setting to zero the histogram values in the open interval between the upper and lower limits of the peak determined in (2) above. The maximum of the new histogram is found, and its position defines the secondary (or minor) peak. This is tested and flagged in the same way as the major peak.

The new maximum represents the largest secondary peak, and will be designated the minor peak in the following. The same validity tests are applied to it as are described above for the major peak.

5 The two peaks are now further tested in relation to one another. These tests may result in a hitherto valid peak being flagged as invalid. In detail the tests are as set out in the detailed description of the algorithm below, but the main new element here is to use the averaged 12 micron brightness temperatures in the peak to refine the discrimination; if the mean brightness temperature for the rightmost peak exceeds that for the other by more than a threshold `MAX_DIF_PEAK_CHAN_1`, the left-hand peak is flagged as invalid. At the same time, if the peaks overlap, the upper limit of the left-most peak is refined by extrapolation, and the lower limit of the rightmost peak (which may for the basis of the threshold determination) is then adjusted upwards to match the extrapolated limit. (This makes for a more conservative detection.)

6 The difference threshold is now set. If only one peak is valid, the difference threshold is set to the lower limit of the valid peak. If both peaks are flagged as valid, that at the lower abscissa is selected and its lower limit is used. If neither peak is valid, all pixels are flagged as cloudy.

7 The correlation property described in paragraph 4 following section (5.5.3.7.9) above is used to validate and, in the case of night-time scenes, refine the threshold just derived. The average 12 micron brightness temperature associated with each histogram box is calculated, using

$$T_{\text{mean}}(k) = \text{sigma}(T_{12})/h(k),$$

where the sum is over all pixels contributing to box k . (The calculation uses the histogram associated with the chosen peak; thus if the minor peak was valid and selected, the histogram after removal of the major peak is used.) The histogram bin corresponding to the maximum brightness temperature is identified, and the slope of the averaged brightness temperature with Bt difference is computed. If this slope exceeds a value `SLOPE_MAX_ALLOWED`, all pixels are flagged as cloudy.

If the histogram represents a night time scene, the average 12 micron BT are used to refine the threshold just derived. If the position of the chosen histogram peak is to the right of the BT maximum, the threshold is increased until the histogram value at the threshold exceeds a given fraction (`1/RATIO_B`) of that at the position of the maximum BT. If the position of the chosen histogram peak is to the left of the BT maximum, the threshold is increased until either the histogram peak is reached, or a bin is found at which



the 12 micron average exceeds the highest average BT less $MAX_DIF_AVE_CHAN - 1$ and the slope is lower than $SLOPE_MAX_ALLOWED$.

- 8 Finally, all so far clear sea pixels having a brightness temperature difference less than the threshold are flagged as cloudy. If the number of clear pixels that remain is less than the threshold $MIN_FOR_11_12_HISTOGRAM$, these pixels are also flagged as cloudy.

9

The detailed algorithm follows.

a) The following steps (b) to (m) are applied to the nadir and oblique views separately. Some of the cloud detection parameters provided in the auxiliary file of Cloud Look-up Table Data differ for the two views, and where applicable those appropriate to the view being processed should be used.

b) The 11um minus 12um brightness temperature differences ($T_{11} - T_{12}$) are calculated for those pixels that have not been flagged by any of the previous cloud tests and are not over land. The histogram of these differences is generated with a bin size of 0.1 K. For each histogram bin the cumulative brightness temperature in the 12 micron channel is calculated for all the pixels that contribute to each bin.

c) If the number of pixels contributing to the histogram is less than the auxiliary value $MIN_FOR_11_12_HISTOGRAM$, all the pixels are flagged as cloud no further processing is applied to this view. (If the scene is less than 512 rows in length, the above threshold is scaled in proportion.)

d) The position of the histogram peak, and the histogram value at the peak, are determined. If possible, the position of the peak of the distribution is derived to sub-box accuracy by fitting a quadratic function to the histogram values at the peak and in the two adjacent boxes, and computing the position and ordinate of the maximum of the quadratic. The peak determined at this step will be termed in the following the major peak.

e) The extent of the peak is determined by finding the position of a local minimum or of an empty histogram box on each side of the peak.

e.1) Starting at the histogram peak and working to the left, inspect each histogram bin until either a bin value of zero or a local minimum lower than $peak_frac_min$ times the peak value is found.

e.2) Check if possible that the trough found is genuine by testing that the histogram value at the local minimum found in e.1 is greater than 10% of the peak value and that the value in the second bin from the minimum in the direction away from the peak is not lower than the local minimum value. If neither of these conditions is true then the search is continued as in e.1. (This step is done only once.)

e.3) The higher limit is found in a similar way but working to the right of the peak (to higher bin numbers).

Note: It is possible, although unlikely, that an upper or lower limit cannot be found at this step. In this case the present implementation will terminate the processing if the current view here.

f) If the number of pixels contributing to the histogram peak is less than the auxiliary value IR_PEAK_MIN , the brightness temperature difference at the histogram peak is less than zero, and the absolute latitude of the image centre is greater than 40 degrees, then the peak is flagged as INVALID. If any of these three conditions do not apply, the peak is flagged as VALID.

Note: the current value of IR_PEAK_MIN is 25000. If the peak value of the histogram is greater than this, the peak will be flagged as VALID.



- g) The average 12um brightness temperature for the pixels that fall in the histogram bin containing the most pixels (the histogram peak) is computed from the cumulative sum calculated at step (b).
- h) If the major peak is VALID, and the solar elevation at the centre of the scene is less than 5 degrees, so that the histogram represents a night-time scene, the following tests are applied to verify that the histogram is not the sum of a clear and a cloudy histogram and hence excessively broad.
- h.1) The width of the histogram at half-height is computed to sub-box accuracy by interpolation, using the two values bracketing the 50% peak values.
 - h.2) A half-width threshold is calculated from the auxiliary parameters appropriate to the view. These parameters define the threshold as a linear function of the (11um - 12um) difference value at the histogram peak.
 - h.3) If the half width found at (h.1) exceeds than half-width threshold from (h.2) the peak is flagged INVALID.
- i) A new histogram is generated by setting to zero the histogram values in the open interval between the upper and lower limits of the major peak. The peak of this histogram, which is a secondary peak of the original histogram, and which in the following will be called the minor peak, is found and tested for validity as follows. If the number of pixels contributing to the new histogram is less than threshold used in Step (c), the minor peak is flagged as INVALID. Otherwise:
- i.1) Steps (d) to (h) are repeated to determine the position of the minor peak and its limits. If the histogram peak is at a difference value less than zero, and the (absolute) latitude of the image centre is greater than 40 degrees, then flag the minor peak as INVALID.
 - i.2) The minor peak is flagged as INVALID if any of the following conditions apply:
 - i.2.1) The difference between the higher and lower limits is 20 cK or less;
 - i.2.2) The major peak is highly likely to be cloudy, and the minor peak is not sufficiently distinct from the major peak:
 - The box number of the minor peak is at a greater difference value than that of the major peak, AND
 - The average 12um BT at the minor peak is greater than that at the major peak by more than threshold *MAX_DIF_PEAK_CHAN_1*, AND
 - the lower limit of the minor histogram is less than the higher limit of the major histogram, AND
 - the value of the major peak at its higher limit is greater than *second_low_fraction* times the minor peak
 - i.2.3) The minor peak is highly likely to be cloudy:
 - The box number of the minor peak is smaller than that of the major peak, AND
 - The average 12um BT at the minor peak is less than that at the major peak by more than threshold *MAX_DIF_PEAK_CHAN_1*
 - i.3) It is possible that there are no empty histogram boxes between the minor and major histogram peaks. In this case, if the major peak is likely to be cloudy, an extrapolation is used to find the intersect of the falling edge of the major peak and the x axis to refine the limits.
 - (i.3.1) If the minor histogram is valid and to the right of the major, the higher limit of the major histogram is adjusted by extrapolation:



(i.3.1.1) If the lower limit of the minor histogram is less than the higher limit of the major, then it is reset to the higher limit of the major histogram.

i.3.1.2) If the difference between the box numbers for the major peak and the high interval for the major peak is greater than 3 then extrapolate using the major peak box number and value, and the box number and value of the box at (high interval minus 1). Specifically, define x_1 , y_1 as the histogram mode abscissa and ordinate respectively;

$x_1 = \text{peak_interval}[\text{MAJOR}]$, $y_1 = \text{peak_value}[\text{MAJOR}]$.

Similarly define

$x_2 = \text{high_interval}[\text{MAJOR}] - 1$, $y_1 = h[\text{high_interval}[\text{MAJOR}] - 1]$.

Then the linearly extrapolated value of x when $y = 0$ is:

$$x = x_1 + y_1(x_2 - x_1)/(y_1 - y_2)$$

This is the new value of $\text{high_interval}[\text{MAJOR}]$.

i.3.1.3) If the difference between the box numbers for the major peak and the high interval for the major peak is not greater than 3 then linearly extrapolate as above but using the major peak box number and value, and the box number and value of the box at high interval.

(i.3.1.4) If the lower limit of the minor histogram is less than the adjusted higher limit of the major, then it is reset to the higher limit of the major histogram, or to the (peak position - 1) if that is smaller.

i.4) If the minor peak is invalid and is to the left of the major histogram then recompute the minor high limit:

(i.4.1) If $(\text{high_limit}[\text{MINOR}] - \text{peak_interval}[\text{MINOR}])$ is greater than 3 histogram bins then re-adjust the minor high limit by linear extrapolation as at Step (i.3.1.2) above using the values of the histogram mode and the bin second to the left of the old limit

(i.4.2) If $(\text{high_limit}[\text{MINOR}] - \text{peak_interval}[\text{MINOR}])$ is not greater than 3 histogram bins then re-adjust the minor high limit by LINEAR extrapolation as at Step (i.3.1.2) above using the values of the histogram mode and the bin immediately adjacent to the old limit

(i.4.3) If $\text{low_limit}[\text{MAJOR}] < \text{high_limit}[\text{MINOR}]$ then reset $\text{low_limit}[\text{MAJOR}]$ to the smaller of $\text{high_limit}[\text{MINOR}]$ or $(\text{peak_interval}[\text{MAJOR}] + \text{peak_interval}[\text{MINOR}])/2$.

(i.5) If the minor peak is to the right of (i.e. at a higher difference value than) the major peak, AND its average 12um brightness temperature is greater by threshold $\text{MAX_DIF_PEAK_CHAN_1}$ than that of the major peak, then flag the major peak as *INVALID*.

j) Choose between the lower limits of the major and the minor peaks, if necessary:

If both peaks are *VALID* then choose the lower value.

If there is only one *VALID* peak then choose its lower limit.

If there is no *VALID* peak then flag all pixels as cloudy and exit.

j.1) For the chosen peak, calculate the difference



$(\text{peak_interval}[\text{CHOSEN}] - \text{low_limit}[\text{CHOSEN}]) * \text{bin_size}$

(where $\text{bin_size} = \text{dT} = 0.1 \text{ K}$). If this quantity is greater than the value of ir_spread appropriate to the view, then replace the previous threshold value by

$$\text{threshold} = \text{peak_interval}[\text{CHOSEN}] - \text{ir_spread}[\text{view}]/\text{bin_size}$$

and omit step (k), otherwise execute step (k).

k) If the lower limit was successfully determined in (i) then check the way the 12um brightness temperature varies with the (11um minus 12um) difference. Under normal conditions, it decreases as the difference increases.

k.1) Average the 12um brightness temperature values for the pixels in each (11um minus 12um) difference histogram box

k.2) Find the (11um minus 12um) difference value that has the highest average 12um brightness temperature.

k.3) Compute the slope of the (average 12um BT) with respect to the (11um minus 12um) difference, at this point.

k.4) If the slope exceeds threshold *SLOPE_MAX_ALLOWED* then flag all pixels as cloudy, and exit.

k.5) At night, use the 12um average BTs to tighten the lower limit, if necessary.

k.5.1) If the peak position from the (11um minus 12um) histogram is at a higher difference value than the difference value from k.2 then, starting with the lower limit calculated above, increase its value until the histogram reaches a certain fraction ($1/\text{RATIO}_B$) of the peak value.

k.5.2) If the peak position from the (11um minus 12um) histogram is at a lower difference value than the difference value from k.2 then

k.5.2.1) Generate slopes of $\text{delta}(12\text{um BT}) / \text{delta}(11\text{um}-12\text{um BT})$ for each non-empty histogram box

k.5.2.2) Starting with the lower limit calculated above, increase its value until, at lower limit, the (12um average plus threshold *MAX_DIF_AVE_CHAN_1*) value exceeds the highest average 12um brightness temperature determined in (k.2), and the value of the slope here is less than threshold *SLOPE_MAX_ALLOWED*

l) Flag as cloudy all formerly clear pixels with a (11um minus 12um) difference value less than the value of lower limit, determined above.

m) If the number of clear pixels remaining is less than a threshold, then flag all the remaining clear pixels as cloudy.

m.1) Calculate the number of clear) pixels remaining;

m.2) If the total is less than threshold value *MIN_FOR_11_12_HISTOGRAM* then flag all clear pixels as cloudy.



5.5.3.6.10 Visible channel cloud test

This test operates on a pixel-by-pixel basis, and applies to daytime pixels only. It is applied to each view separately.

The test makes use of an NDVI-based surface classification from unpublished work at RAL by A.D. Stevens. The classification scheme uses two NDVI-like indices to classify individual pixels on the basis of pre-assigned criteria. The NDVI may be defined as

$$NDVI = (R87 - R67)/(R87+R67),$$

where R87 and R67 are the calibrated reflectances in the 0.87 and 0.67 micron channels respectively. In addition to the conventional NDVI defined above a second index may be defined making use of the 0.55 micron channel reflectance R55:

$$NDI2 = (R67 - R55)/(R67 + R55)$$

If these two indices are calculated for each pixel, the pixel can be plotted on a graph of NDVI versus NDI2. Such a plot defines an NDVI space within which pixels of different surface types form clusters, and by identifying into which cluster a pixel falls, the surface type at the pixel can be determined.

The clusters have been defined by empirically dividing the NDVI space into a series of polygons, each of which represents a particular surface type. Figure 5-12 shows this subdivision. The classification defines 12 surface types (numbered 0 to 11). Note that the figure shows, in the centre, a pure cloud type (Type 3) together with 4 mixed types incorporating some cloud (Types 1, 2 9 and 10).

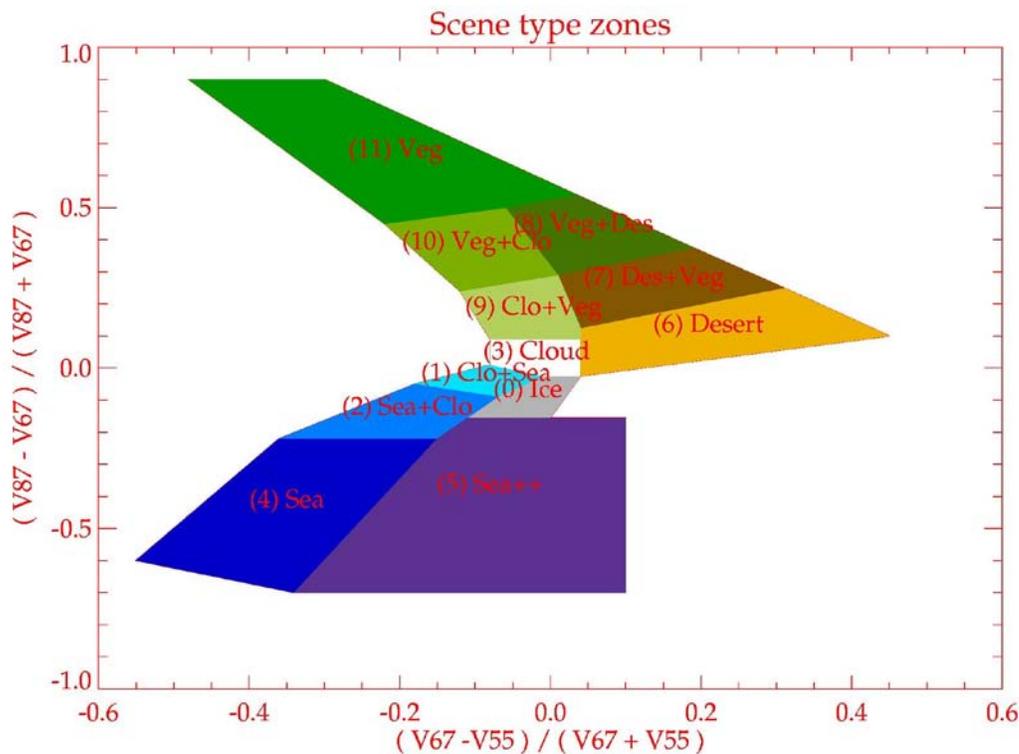


Figure 5-12 Surface type zones according to the classification discussed in the text.



Thus the test proceeds by calculating both indices for each valid pixel and determining the polygonal zone into which it falls. The cloud flag is then set if the pixel falls into one of the zones designated cloudy. The polygonal zones are defined by the co-ordinates [NDVI, NDV2] of its vertices.

Auxiliary parameters used by the visible channel cloud test are defined in the general parameters, zone vertices and zone definitions table in RD-11. These provide the information on the positions of the zones. There is provision for one set of vertices for each of 3 air mass ranges ($v=na$, int , ob (near-nadir, intermediate, oblique)).

N_SIDES	Number of vertices (or sides) per zone	
N_ZONES	Number of defined zones	
Vertex_index[j_ver]	Vertex index of vertex j_ver	$j_ver = 0, \dots, NSIDES - 1$
Zone_index[i_zone]	Zone index of zone i_zone	$i_zone = 0, \dots, N_ZONE - 1$
Vetex_id[j_ver, i_zone]	Vertex identifier	$j_ver = 0, \dots, NSIDES - 1$, $i_zone = 0, \dots, N_ZONE - 1$
X_v[j_ver, i_zone]	X co-ordinate of vertex j_ver in zone i_zone	$j_ver = 0, \dots, NSIDES - 1$ $i_zone = 0, \dots, N_ZONE - 1$
Y_v[j_ver, i_zone]	Y co-ordinate of vertex j_ver in zone i_zone	$j_ver = 0, \dots, NSIDES - 1$ $i_zone = 0, \dots, N_ZONE - 1$

Modifications to the AATSR test: This is a single pixel test based on a surface classification. In AATSR processing the same classification LUT is used for both oblique and nadir views, so to this extent the test can be carried over unchanged. In principle slightly different tables could be used for the nadir and inclined view, and if this idea were adopted, we might also divide the nadir swath into zones as above by across-track limits X_1, X_2 such that:

if $-X_1 < X < X_1$ the AATSR nadir view definitions are used;
if $X_1 < \text{abs}(X) < X_2$, an intermediate set of definitions are used;
of $X_2 < \text{abs}(X)$, the inclined view definitions are used.

The following steps are applied for each view ($v = n \mid f$).

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

$\langle \text{view} \rangle_solar_elevation(i, 0) \geq 5.0$ or
 $\langle \text{view} \rangle_solar_elevation(i, \text{max } j) \geq 5.0$,

so excluding night-time data.

For each day-time pixel j identified above for which the visible channel reflectance values are valid, i.e. for which $I(ch, v; i, j) > 0$ for $ch = v870, v670, v555$ calculate the normalised difference indices as follows:

$$NDVI = (I(v870, v; i, j) - I(v670, v; i, j)) / (I(v870, v; i, j) + I(v670, v; i, j))$$

$$NDI2 = (I(v670, v; i, j) - I(v555, v; i, j)) / (I(v670, v; i, j) + I(v555, v; i, j))$$

Perform the following procedure to identify within which cluster defined by $NDVI$ and $NDV2$ the pixel falls.

For each zone $i_zone = 0, N_ZONE - 1$

Extract array of vertices:

$Vertex[j_ver] = vertex_id[j_ver, i_zone], j_ver = 0, N_SIDES - 1$

Define arrays (X_ar, Y_ar) of dimension $N_SIDES + 1$ and extract the vertex co-ordinates into them:

For each vertex $j_ver = 0, N_SIDES - 1$



$$X_arr[k] = X(k)$$

$$Y_arr[k] = Y(k)$$

End for

$$X_arr[N_SIDES] = X[0]$$

$$Y_arr[N_SIDES] = Y[0]$$

Identify whether the point *PX*, *PY* lies within this zone.

FLAG = *TRUE*

PX = *NDI2*

PY = *NDVI*

For *k* = 0, *N_SIDES* – 1

$$SX = X_arr[j_ver+ 1] - X[j_ver]$$

$$SY = Y_arr[kj_ver+ 1] - Y[j_ver]$$

$$QX = PX - X[j_ver]$$

$$QY = PY - Y[j_ver]$$

Calculate the vector cross product *Q* × *S*:

$$ZZ = (QX \times SY - QY \times SX)$$

FLAG = *FLAG* AND (*ZZ* ≥ 0)

If *FLAG* = *FALSE* then exit loop; the point is not in this zone

End for

If *FLAG* = *TRUE*, exit loop ; we have found the zone

End for

The table of zones will only contain those regions that are associated with cloud. Thus the value of *FLAG* at this point tells whether or not the pixel is cloudy.

If *FLAG* = *TRUE* and the pixel is a land pixel for which the land flag (provided by the confidence parameter in the global flags) is set then flag the pixel as cloudy by setting the cloud flag *vis_test* for each view.

(Implementation note: although this algorithm in theory works over sea as well as land, use over sea is disabled for the time being. This is done in this step rather than by applying the test as a whole to land pixels only so that it can be easily enabled for sea pixels if desired at some time in the future.)

5.5.3.6.11 1.375 micron threshold test

It is proposed that a simple test based on the new 1.375 micron channel be used to detect cirrus cloud. The present test is based on a fixed reflectance threshold, similar to that defined for MODIS [RD-12].

This is a threshold test applied on a pixel by pixel basis to the 1.375 micron channel data. The physical basis of the test is that water vapour absorption in this band is very strong, so that if the atmosphere contains more than about 1 cm of total precipitable water in a column, no radiation from the surface will reach the satellite. Except in the driest regions therefore, any signal received in this channel will be the result of reflection or scattering by cloud. The basic test is then that if the observed reflectance exceeds a threshold (corrected for solar incidence angle), cloud is flagged.

The test is only applicable to daytime pixels, which for consistency with other tests are defined by a solar elevation exceeding 5 degrees, but it is valid over both land and sea. It is not applied if the surface elevation exceeds 2000 m, where precipitable water may be low, and it may be invalid in polar regions for the same reason.

The following steps are applied for each view (*v* = *nadir* (*NA*) | *oblique* (*OB*)).

Retrieve the threshold for the view *v*, *R137_THRESHOLD_v* from the auxiliary data file of cloud look-up table values.

For each scan *i*, retrieve the solar elevation at each end of the scan, as in Section 5.5.3.7.11 above. The test is only performed if these are not less than 5°, so excluding night-time data.



A linear interpolation may be used to determine the solar elevation at intermediate points, again as in Section 5.5.3.7.11 above. Denote this by *sol_elev*.

Extract the surface elevation. Only proceed with the test if the solar elevation exceeds 5 degrees and the elevation is less than 2000 m.

The solar zenith angle in radians is

$$z = \pi(90 - \text{sol_elev})/180.0.$$

The calibrated reflectance at 1.37 microns, corrected to normal solar incidence, is then

$$R137 = (I(v137, v; i, j) * \sec(z))$$

If ($R137 > R137_THRESHOLD$), the pixel is considered cloudy. Flag the pixel accordingly using the *thresh_test* cloud word.

Note: although the test is described here with a fixed threshold, it may be necessary to provide a look-up table of threshold values as functions of solar zenith and viewing angles (TBD).

5.5.3.6.12 2.25 micron histogram test

It is proposed that a dynamic threshold test similar to the 1.6 micron test but using the 2.25 micron pixel may be defined. The test is performed as above, but substituting the calibrated reflectance in the 2.25 micron channel, along with a different dedicated look-up table of parameters.

5.5.3.6.13 Snow-covered surface test

This test operates on a pixel-by-pixel basis, and applies to daytime pixels only. It is applied to each view separately.

Modifications to the AATSR test: This test can be carried over unchanged. It is not strictly a cloud test, and the result is not masked into the overall cloud flag. Note that MODIS experience suggests that it may be unreliable at high incidence angles.

The following steps are applied for each view ($v = n | f$).

For each scan *i*, retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5°: i.e. if

$$\begin{aligned} <view>_solar_elevation(i, 0) \geq 5.0 \text{ or} \\ <view>_band_centre_solar_elevation(i, \max j) \geq 5.0, \end{aligned}$$

so excluding night-time data.

For each day-time pixel *j* identified in step (5.22.3-114) above for which the 1.6, 0.87 and 0.55 micron channel reflectance values are valid, i.e. for which $I(ch, v; i, j) > 0$ for $ch = v16, v870, v555$ calculate the normalised difference snow index *NDSI* as follows:

$$NDSI = (I(v555, v; i, j) - I(v16, v; i, j)) / (I(v555, v; i, j) + I(v16, v; i, j))$$

The across-track band number for each pixel in the regridded image is given by

$$\text{band}(j) = 0 \text{ IF } j < 6$$



$\text{band}(j) = \text{integer part of } (j - 6) / 50 \quad \text{IF } 6 \leq j < 506$

$\text{band}(j) = 9 \quad \text{IF } j \geq 506$

where j is the pixel across track index.

A linear interpolation may be used to determine the solar elevation.

$w = \text{float}(j - 6) / 50.0 - \text{band}(j)$

$\text{sol_elev} = (1.0 - w) * \text{nadir_band_edge_solar_elevation}(i, \text{band}(j)) +$
 $w * \text{nadir_band_edge_solar_elevation}(i, \text{band}(j) + 1)$

The solar zenith angle in radians is

$z = \pi(90 - \text{sol_elev}) / 180.0.$

The calibrated reflectance at 0.87 microns, corrected to normal solar incidence, is then

$R87 = (I(v870, v; i, j) * \sec(z))$

Extract the 11 micron brightness temperature:

$T11 = I(ir11, v; i, j)$

Convert the units of the NDSI threshold:

$NDSI_THRESH = NDSI_THRESH / 10000.$

If $T11 < T11_THRESH$ and ($R87 > R87_THRESH$ and $NDSI > NDSI_THRESH$), the pixel is considered snow-covered.



5.5.3.7 Convert Solar Channel Reflectances to Radiance

So far all processing relating to the S1 to S6 channels has been carried out using calibrated reflectances. It is necessary at the output stage to convert all such reflectance values to radiance by multiplication by the appropriate scaling factors representing the solar irradiance.

Suppose the solar irradiance at wavelength λ is $E_0(\lambda)$, such that the radiant flux from the sun falling normally on an area dA in wavelength range $d\lambda$ is

$$d\phi = E_0(\lambda)dAd\lambda$$

During calibration, the scene pixels in the solar channels are in effect compared with the reflected energy from a diffuser of known reflectance R_0 , normally illuminated by the sun. The energy reflected from the diffuser into solid angle $d\Omega$ is then

$$d\phi = R_0E_0dAd\lambda d\Omega / \pi$$

The equivalent spectral radiance L_0 is $\frac{1}{\pi} R_0E_0(\lambda)$. Thus if a the calibrated reflectance of a pixel is R , the

equivalent spectral radiance at the detector is $\frac{1}{\pi} R_{scene}E_0(\lambda)$. In practice what us measured by SLSTR is

the integral of this quantity over the spectral profile of the channel $sp(\lambda)$. It follows that the calibrated quantity is the mean spectral radiance defined by

$$\frac{R_{scene}}{\pi} \int E_0(\lambda)sp(\lambda)d\lambda / \int sp(\lambda)d\lambda$$

We define

$$E_{0,\lambda} = \int E_0(\lambda)sp(\lambda)d\lambda / \int sp(\lambda)d\lambda$$

to be the solar irradiance weighted by the SLSTR instrument passband. It is noted that $E_{0,\lambda}$ is a function of view and detector number. Then the conversion from the 'reflectance' computed in step 5.5.3 of section 5.4.3.4 to radiance for a given pixel is simply

$$L_{scene} = R_{scene}E_{0,\lambda}/\pi$$

where R_{scene} is the calibrated reflectance of the channel. The seasonally adjusted values of $E_{0,\lambda}$ for each SLSTR channel are output from the visible calibration algorithm (section 5.4.3.4).



5.5.3.8 Meteo annotations

Models do not in general provide parameter data sets contemporary and co-located with the SLSTR samples. Then, temporal and spatial interpolations are necessary.

Prior to any processing, a checking step is performed:

1. Test of the availability of meteorological data at a time preceding and following the mid-exposure product time
2. Evaluation of the confidence in the temporal interpolation of meteo data. For this purpose a Product Confidence Data (PCD) is set according to the length n of the time interval, separating the two meteo data files, relative to a reference length N_{ref} (corresponding to the usual meteo time sampling, of 6 hours). For example, the formula used to compute this PCD could be:

$$PCD_{meteo_temporal_interpolation} = \left[\text{int}(n) / N_{ref} \right] - 1 \quad (1)$$

At first, meteorological data are linearly interpolated between their values available before and after the mid-exposure product time. It enables to provide meteo data sampled on the original environment spatial grid at the mid-exposure time. Secondly, a bi-linear interpolation is applied to this grid, in order to compute meteo data on the tie points grid chosen, following these steps:

1. coordinates of the 4 environment spatial grid nodes enclosing each tie point are computed,
2. meteo parameters are extracted at these 4 grid points,
3. their values are spatially interpolated at the tie point location by a bi-linear method

Finally, re-sampled meteo data are copied in the product annotation.

APPENDIX A – BREAKPOINT TABLE

The following data shall be used as breakpoints in the testing of the Level 1b process. The final column indicates the accuracy with which the data should be verified against the output of the reference processor. Note that the Parameter Identifiers in the first column are historical, and should be updated to provide an unambiguous cross-reference to the DPM.

Note: In the table, ‘Generally exact’ relates to flags or quantities of type integer, and indicates that test results should agree exactly with the reference processor in the majority of cases, but that a small number (TBD) of discrepancies may acceptable owing to differences in machine precision.

Similar verification accuracies apply to the Level 1B Product Measurement Data sets. The relevant verification accuracies are:

Brightness temperature parameters	0.01 K
Visible Channel reflectances	0.01 %
Confidence and Cloud/land flags	Generally exact
GBTR ADS Quantities	As corresponding quantities in Table 3-1.

Table A-1: Breakpoints

Parameter ID	Name	Verification Accuracy
From Stages 1 to 5:		
L1B-INT-003	auxiliary_data_validation_result[i]	exact
L1B-INT-004	converted_auxiliary_data[i]	dependent upon datum type
L1B-INT-005	unpacked_auxiliary_data[i]	exact
From Stage 6 (Science Data Processing)		
L1B-INT-080	unpacked.pixels.nadir[MAX_NADIR_PIXELS]	exact
L1B-INT-081	unpacked.pixels.along-track[MAX_ALONG_TRACK_PIXELS]	exact
L1B-INT-082	unpacked.pixels.plus_bb[MAX_PXBB_PIXELS]	exact
L1B-INT-083	unpacked.pixels.minus_bb[MAX_MXBB_PIXELS]	exact
L1B-INT-084	unpacked.pixels.viscal[MAX_VISCAL_PIXELS]	exact
From Stage 7 (IR Channel Calibration)		
L1B-INT-006	calibration_invalid[channel]	exact
L1B-INT-010	gain[parity][channel] (channel[,] slope)	1 part in 1e6
L1B-INT-011	offset[parity][channel] (channel[,] intercept)	1 part in 1e6
From Stage 8 (Visible/SWIR Channel Calibration)		
L1B-INT-410 - 438	Visible/SWIR Calibration Product Parameters	dependent upon datum type
From Stage 9 (Satellite Time Calibration)		
L1B-INT-400	source_packet_ut_time(s)	1e-12 days (1 microsecond)
From Stages 12 (Geolocate Pixels), 14 (Interpolate Pixel Positions)		
L1B-INT-060	nadir scan pixel latitude	1 part in 1e6
L1B-INT-061	nadir scan pixel longitude	1 part in 1e6
L1B-INT-062	along-track scan pixel latitude	1 part in 1e6
L1B-INT-063	along-track scan pixel longitude	1 part in 1e6
From Stages 13 (Calculate Pixel x and y co-ordinates), 14 (Interpolate Pixel Positions)		
L1B-INT-064	nadir scan x coordinate (source packet nadir pixel x coords)	1 m
L1B-INT-065	nadir scan y coordinate (source packet nadir pixel y coords)	10 m
L1B-INT-066	along-track scan x coordinate (source packet along-track pixel x coords)	1 m
L1B-INT-067	along-track scan y coordinate (source packet along-track pixel y coords)	10 m
From Stage 18 (Signal Calibration)		
L1B-INT-087	calibrated.pixels.nadir[MAX_NADIR_PIXELS], infra-red /fire channels	0.01K
L1B-INT-088	calibrated.pixels.inclined[MAX_ALONG_TRACK_PIXELS], infra-red/fire channels	0.01K



Parameter ID	Name	Verification Accuracy
L1B-INT-089	calibrated.pixels.nadir[MAX_NADIR_PIXELS], visible/SWIR channels	0.01%
L1B-INT-090	calibrated.pixels.inclined[MAX_ALONG_TRACK_PIXELS], visible/SWIR channels	0.01%
From Stage 15 (Solar and Viewing Angles)		
L1B-INT-120 - 127	nadir_band_edge_<solar and viewing angles>	1 part in 1e6
L1B-INT-140 - 147	along_track_band_edge_<solar and viewing angles>	1 part in 1e6
From Stage 19 (Regrid Pixels).		
L1B-INT-101	regridded nadir ir12 Brightness Temp.	0.01K
L1B-INT-102	regridded nadir ir11 Brightness Temp.	0.01K
L1B-INT-103	regridded nadir ir37 Brightness Temp.	0.01K
L1B-INT-102A	regridded nadir ir11 Fire Channel Brightness Temp.	0.01K
L1B-INT-103A	regridded nadir ir37 Fire Channel Brightness Temp.	0.01K
L1B-INT-104	regridded nadir v2.25 Reflectance	0.01%
L1B-INT-104A	regridded nadir v1.6 Reflectance	0.01%
L1B-INT-104B	regridded nadir v1.375 Reflectance	0.01%
L1B-INT-105	regridded nadir v870 Reflectance	0.01%
L1B-INT-106	regridded nadir v670 Reflectance	0.01%
L1B-INT-107	regridded nadir v555 Reflectance	0.01%
L1B-INT-108	nadir x offset(i, j)	5 m
L1B-INT-109	nadir y offset(i, j)	5 m
L1B-INT-111	regridded along-track ir12 Brightness Temp.	0.01K
L1B-INT-112	regridded along-track ir11 Brightness Temp.	0.01K
L1B-INT-113	regridded along-track ir37 Brightness Temp.	0.01K
L1B-INT-112A	regridded along-track ir11 Fire Channel Brightness Temp.	0.01K
L1B-INT-113A	regridded along-track ir37 Fire Channel Brightness Temp.	0.01K
L1B-INT-114	regridded along-track v2.25 Reflectance	0.01%
L1B-INT-114A	regridded along-track v1.6 Reflectance	0.01%
L1B-INT-114B	regridded along-track v1.375 Reflectance	0.01%
L1B-INT-115	regridded along-track v870 Reflectance	0.01%
L1B-INT-116	regridded along-track v670 Reflectance	0.01%
L1B-INT-117	regridded along-track v555 Reflectance	0.01%
L1B-INT-118	along-track x offset(i, j)	5 m
L1B-INT-119	along-track y offset(i, j)	5 m
L1B-INT-134	nadir view instrument scan number	Generally exact (See note)
L1B-INT-135	nadir view instrument pixel number	Generally exact (See note)
L1B-INT-154	along-track view instrument scan number	Generally exact (See note)
L1B-INT-155	along-track view instrument pixel number	Generally exact (See note)
From Stage 20 (Cosmetic Fill)		
L1B-INT-100	nadir fill state(i, j)	Generally exact (See note)
L1B-INT-110	along-track fill state(i, j)	Generally exact (See note)
From Stage 21 (Determine Image Pixel Positions)		
L1B-INT-160	image_latitude(i, j)	1 part in 1e6
L1B-INT-161	image_longitude(i, j)	1 part in 1e6
From Stage 22 (Determine Land/Sea Flag)		
L1B-INT-232	nadir land flag	Generally exact (See note)
L1B-INT-248	along-track land flag	Generally exact (See note)
From Stage 23 (Cloud Clearing)		
L1B-INT-233 - 244	nadir cloud flags	Generally exact (See note)
L1B-INT-249 - 260	along-track cloud flags	Generally exact (See note)



Change sheet for SLSTR L0/L1 ATBD

Prepared by Caroline Cox (RAL)

Original Comments – Version 1 – 13-Sep-2010

1. Section 5.3.1., p61

Email comment (3) from IM, 31/08/10

section 5.3.1. and 5.3.3.1.: you are talking about 8 detectors even for the visible channels, do you confirm that only 4 are used for visible channels ?

Proposed change to second sentence in the third paragraph of this section

Text was:

'We assume that the mechanical alignment is sufficiently good that the same set of position elements will apply to each channel group (2 element or 8 element)'

Now:

'We assume that the mechanical alignment is sufficiently good that the same set of position elements will apply to each channel group (2 element (TIR), 4 element (VIS) or 8 element (SWIR))'

2. Section 5.3.2.2, p62

Email comment (4) from IM, 31/08/10

section 5.3.2.2.: the sampling of the data to be used at several stages of the geolocation process is not clear:

* stage 11: During our meeting in march, we talked about doing this interpolation at every pixel time. Besides, this is what is done in step 5.3.3.5. Do you confirm that the navigation and attitude should be interpolated at every pixel time and not only at every scan time ?

Proposed Change to stage 11, text

Text was:

'This Stage uses the CFI with the NAVAT to determine the position and orientation of the platform at the time of each scan. '

Now:

'This Stage uses the CFI with the NAVAT to determine the position and orientation of the platform at the time of each pixel of every scan. '



3. Section 5.3.2.2, p62

Email comment (4) from IM, 31/08/10

* stage 12: During our meeting in march, we agreed to do the orthogeolocation process for every instrument pixel,

Proposed Change to stage 12, text

Text was:

'The direction of the line of sight to each tie point pixel is calculated, and used to determine the geodetic latitude and longitude, on the reference ellipsoid, of the tie point.'

Now deleted.

4. Section 5.3.2.2, p62

Email comment (4) from IM, 31/08/10

* stage 13: In SY-4 the (x,y) coordinates are computed for every pixels, do you confirm that this process should be led for every pixels ?

Proposed change to remove reference to tie point from stage 13 text

Text was:

'The x-y (across-track and along-track) coordinates of each tie point pixel are derived from the pixel latitude and longitude.'

Now:

'The x-y (across-track and along-track) coordinates of each pixel are derived from the pixel latitude and longitude.'

5. Section 5.3.2.2, p62

Email comment (4) from IM, 31/08/10

* stage 14: Do you confirm that solar and viewing angles are computed for every pixels (in particular for cloud tests) and then sub-sampled to be stored in the SLSTR L1b product ?

Text was:

'The angles are calculated using the CFI at a series of tie points around the scan at increments of 50 km (TBC) in x.'

Now:



'The angles are calculated using the CFI at every pixel and then sub-sampled at increments of 50 km (TBC) in x.'

6. Section 5.3.3.1, p64

Email comment (3) from IM, 31/08/10

section 5.3.1. and 5.3.3.1.: you are talking about 8 detectors even for the visible channels, do you confirm that only 4 are used for visible channels ?

Text was:

'k Detector element index. k = 0, 1 for the thermal channels, k=0, ... 7 for the solar channels.'

Now:

'k Detector element index. k = 0, 1 for the thermal channels, k=0, 4 for the visible channels, k=0, ... 7 for the SWIR channels.'

7. Section 5.3.3.3, p67

Email comment (6) from IM, 31/08/10

section 5.3.3.3.: Do you confirm that the time is computed on four grids for each view as said in SY-4 section 6.7.1.5.6.1 ? The processing for the time of TDI pixels is missing in this section, could you please confirm that this should be the mean of the times of stripe A and stripe B ?

Text was:

'This step is performed for every ISP.'

Now:

'This step is performed for every ISP. The time is computed on four grids, the TIR grid, stripe A, stripe B and TDI grid where the TDI time is simply the average of the A and B grid times.'

8. Section 5.3.3.3.2, p68

Email comment (1) from IM, 31/08/10

* TDI grid is not handled in the processing. Thus, for the time being I did not include it into the geolocation process, since I guess that (lat,lon) and (x,y) coordinates will be deduced from stripe A and B, but could you please provide the exact processing to be applied to compute the TDI grid ?

AND

Email comment (7) from IM, 31/08/10

section 5.3.3.3.2: For the time being the special case of channel F1 is not taken into account in SY-4, only 1 grid per view is foreseen for all channels provided at 1km resolution. For the time being I have followed the SY-4 definition, which induces that pixel time of F1 is the same as the one of other thermal channels. Do you agree with this approach ?

Response:



Addition of TDI pixel time and a statement that the F1 channel uses the same time as the TIR channels. At the end of the section text

Text was:

*'for SWIR stripe B channels,
offset = PIX10SYNC/4 + PIX10SYNC/2 for the nadir view and
offset = PIX10SYNC/4 - PIX10SYNC/2 for the oblique view TBC!!*

*In summary, there are therefore 4 different sets of pixel timings for SLSTR:
t_pix_10 pixel time associated with TIR channels S7-S9 and F2
t_pix_10F1 pixel time associated with F1
t_pix_05A pixel time associated with VIS channels S1-S3 and A stripe of SWIR channels S4-S6
t_pix_05B pixel time associated with B stripe of SWIR channels S4-S6'*

Now:

*'for SWIR stripe B channels,
offset = PIX10SYNC/4 + PIX10SYNC/2 for the nadir view and
offset = PIX10SYNC/4 - PIX10SYNC/2 for the oblique view*

*The TDI time grid is calculated as
 $t_pix_TDI = (t_pix_05A + t_pix_05_B) / 2$*

*In summary, there are therefore 5 different sets of pixel timings for SLSTR:
t_pix_10 pixel time associated with TIR channels S7-S9 and F2
t_pix_10F1 pixel time associated with F1
t_pix_05A pixel time associated with VIS channels S1-S3 and A stripe of SWIR channels S4-S6
t_pix_05B pixel time associated with B stripe of SWIR channels S4-S6
t_pix_TDI pixel time associated with TDI of SWIR channels*

However the F1 channel is assigned the t_pix_10 pixel timings as the difference is negligible'

9. Section 5.3.3.4, step 5.3.3.4.1, p69

Email comment (8) from IM, 31/08/10

o deltaT is not directly available from ADF, so I took cycle_period, cycles_per_tie_point, and scans_per_cycle values provided in L1b processor configuration ADF following this formula: $\Delta T = \text{cycles_per_tie_point} * \text{scans_per_cycle} * (\text{cycle_period} / 2.0)$. Do you agree with this approach ?

Text was:

'The time step ΔT is chosen to correspond to one granule.'



Now:

'The time step ΔT is chosen to correspond to one granule and calculated as follows:

$$\Delta T = \text{cycles_per_tie_point} * \text{scans_per_cycle} * (\text{cycle_period} / 2.0)$$

10. Section 5.3.3.6, p73

Email comment (9) from IM, 31/08/10

* Do you confirm that K' angle, used to define the transformation matrix Mab from the scan reference frame to the instrument one, is the scan_inclination_nadir defined in Geometry ADF (SY-4 section 6.8.3.6) ?

Proposed Change to text

'K' is the scan_inclination_nadir defined in Geometry ADF'

Is to be inserted after

'The matrix Mab enables one to go from the scan reference frame to the instrument reference frame.'

11. Section 5.3.3.6, p74

Email comment (9) from IM, 31/08/10

* Do you confirm that Kv angle, used to define the matrix Mcm, is the scan_cone_angle defined in Geometry ADF (SY-4 section 6.8.3.6) ?

Proposed Change to text

'Kv is the scan_cone_angle defined in Geometry ADF'

Is to be inserted after

'The index v identifies the view (v = n | a = nadir | along-track)'

12. Section 5.3.3.6, p75

Email comment (9) from IM, 31/08/10

* step 5.3.3.6.2.:

o The switch between characterized misalignment matrices and the geometric calibration matrix should be added in the L1b processor configuration parameters

NO CHANGE TO DOCUMENT. This is an implementation issue.

13. Section 5.3.3.6, p75

Email comment (9) from IM, 31/08/10



o for the computation of quaternions, the interpolation to be used should be SLERP (according to TAS-F) instead of the classic one proposed in the ATBD

text was:

*'3. Computation of the quaternion $Q_v(X_v)$ corresponding to X_v :
a. Classic (e.g. cubic) interpolation at location X_v in the LUT obtained at step 1. '*

Now:

*'3. Computation of the quaternion $Q_v(X_v)$ corresponding to X_v :
a. SLERP interpolation at location X_v in the LUT obtained at step 1. '*

14. Section 5.3.3.7, p79

Email comment (10) from IM, 31/08/10

* the first paragraph is not clear ... do you confirm that the (c,y) coordinates computation should be led at every pixels of every scans, i.e. at full resolution as suggested by SY-4 section 6.7.1.5.5.7 ?

Text was:

'The following steps (described in section 4.4.4.10) are carried out for the same set of tie points as defined in Req. 5.12-1 on every INT_S'th scan st. The pixel ortho-geolocated latitudes and longitudes have been calculated for specific tie points on the scan: on the nadir scan these points are $P_t^o = 0, 10, 20, 30, \dots, MAX_NADIR_PIXELS - 1$ eq 5.3-61 and for the oblique scan $P_t^o = 0, 10, 20, 30, \dots, MAX_ALONG_TRACK_PIXELS - 1$ eq 5.3-62'

Now:

'The following steps (described in section 4.4.4.10) are carried out for every pixel. The ortho-geolocated latitudes and longitudes have been calculated for every pixel on the nadir scan as $P_t^o = 0, 1, 2, 3, \dots, MAX_NADIR_PIXELS - 1$ eq 5.3-61 and for the oblique scan $P_t^o = 0, 1, 2, 3, \dots, MAX_ALONG_TRACK_PIXELS - 1$ eq 5.3-62'

16: Section 5.4.3 , p89

Email comment (1) from IM, 01/09/10

What is the range of Target_ID? [0, 7] or [A0h, D1h] as in table 6-6 in SY-4 document or [00h, FEh] as in table 6-12 in SY-4 document?

According to the IMDD values from 00h to FEh are used. Table 6-12 refers to the allowed range, whereas 6-6 refers to the allocated values.

NO CHANGE MADE TO DOCUMENT.



17: Section 5.4.3.1.6, p89

Email comment (2) from IM, 01/09/10

* We have to distinguish items which can be converted and the others for which no conversion is sensible. How is this distinction drawing? When function_ID equals 'f0', the conversion function is a simple float function. Does it mean that 'function_ID = = f0' is the distinction mark for unconverted data?

Text was:

'f0: float(Z)'

Now:

'If a conversion is not necessary for a parameter, it has a function ID=f0

f0: float(Z)'

18: Section 5.4.3.1.6, p89

Email comment (2) from IM, 01/09/10

* What is the range of function_ID: [0, 20] as in the section 6.1.1.1.1 in SY-4 document, or [f0, f20] as in the section 5.4.3.1.6 in the ATBD document?

Both. NO CHANGE MADE TO DOCUMENT

19: Section 5.4.3.1.6 p91

Email comment (2) from IM, 01/09/10

* For function_ID = 'f20', the conversion function is an interpolation from the 2xN parameter LUT provided in the ADF. Could you precise which type of interpolation has to be used? I implemented a linear interpolation, is it ok?

Text was:

'The conversion from Z to anc_conv_data is performed by an interpolation function.'

Now:

'The conversion from Z to anc_conv_data is performed by a polynomial interpolation function.'

20: Section 5.4.3.1.7, p91

Email comment (3) from IM, 01/09/10

* The "general limit checks" is applied to all converted items for which surveillance limits are specified. What is the default value of surveillance limits for item not concerned by this "general limit checks"?

If no limit is specified then no check is made. NO CHANGE MADE TO DOCUMENT



21: Section 5.4.3.1.7, p91

Email comment (3) from IM, 01/09/10

* The “black Body over range check” is performed if an over-range flag is associated with each black-body temperature transducer. Could you precise where can be found this over-range flag?

At the moment there is no document describing the TM so it is not defined at the moment. I would rely on the surveillance checks and limits checks to see which sensors are “overrange”.
NO CHANGE MADE TO DOCUMENT

22: Section 5.4.3.1.7.2, p93

Email comment (3) from IM, 01/09/10

* Could you precise the term “heated black body” and the parameter determining its nature (BB1 or BB2)?

Was:

‘If the heated black body is BB1 then’

Now:

‘The heated black body is determined by comparing the temperatures of BB1 and BB2. The black body with the highest temperature is then the heated one.

If the heated black body is BB1 then’

23: Section 5.4.3.1.7.2, p93-94

Email comment (3) from IM, 01/09/10

* In the “Check each black body sensor temperature” section, the set of valid temperature values is dividing into two subsets K1 and K2, based on their geometric layout. Could you precise how this division is made?

Text was:

‘We divide the set of valid temperature values into two subsets K1, K2 on the basis of their geometric layout (TBD).’

Now:

‘We divide the set of valid temperature values into two subsets K1, K2 on the basis of their geometric layout. The thermometers should be arranged so that there is a group of 3 near the centre of the black body baseplate, two towards the edge and one on the baffle. To get the correct weighting we need to group the three on the centre together and the two on the edge. We do not use the baffle sensor in the average. The precise location of each sensor is TBD.’

24: Section 5.4.3.1.7.2, p94



Text was:

'Step 5.5.3-7-1'

Now:

'Step 1'

25: Section 5.4.3.1.7.2, p95

Text was:

'Step 5.5.3-7-2'

Now:

'Step 2'

26: Section 5.4.3.1.7.2, p96

Email comment (3) from IM, 01/09/10

* This test has to be done for each scanner. Could you precise this term? Does it refer to the position MSB, MIDB and LSB?

Text was:

'The test is performed over the nadir and oblique views for each SCANSYNC period and for each scanner as follows'

Now:

'The test is performed over the nadir and oblique views for each SCANSYNC period as follows'

28: Section 5.4.3.1.7.2, p98

Email comment (4) from IM, 01/09/10

* Concerning the scan Time Inconsistency checks, on which ISP file must we do this test?

Text was:

'This checks that the time in the ISP is consistent with the scan count'

Now:

'This checks that the time in the SCAN TIME ISP is consistent with the scan count'



29: Section 5.4.3.2, p99

Email comment (5) from IM, 01/09/10

In the science Data Processing (Stage 6, section 5.4.3.2) could you confirm that, for thermal IR and fire channels, there is no indexation by the parameter t (i.e. the index to the PIX05SYNC interval within the sample was measured)?//

Correct because the IR channels are sampled every PIX10SYNC us.
NO CHANGE MADE TO DOCUMENT

30: Section 5.4.3.3 step 5, p103

On page 103 (Section 5.4.3.3 step 5), the formula to calculate the radiometric offset should read

$$\text{offset } [ch,view ,t,k] = \frac{L_true_{BB1}(ch)\overline{C}_{bb}[ch,ID_2,t,k] - L_true_{BB2}(ch)\overline{C}_{bb}[ch,ID_1,t,k]}{\overline{C}_{bb}[ch,ID_2,t,k] - \overline{C}_{bb}[ch,ID_1,t,k]}$$



Comments added in version 2 of change sheet - 15/09/2010

1. Section 5.3.3.4, p69

Email IM comment (4), 01/00/10

o K is not provided in any ADF, could you please clarify where this auxiliary constant could be found ?

K has been added to SY4 and given the name additional_scan_constant. Text was:

'K is a constant offset that defines the number of samples prior to the ascending node prior to the start of the table, or output product start time if different.'

Now:

*'K is a constant offset that defines the number of samples prior to the ascending node prior to the start of the table, or output product start time if different. **K=additional_scan_constant in the L1b processor ADF.**'*

2. Section 5.3.3.4, p69

Typo corrected. Text was:

's(ak) Along-track distance of sample k. Again, not to be confused with the scan trace index defined in section 4.2.3.'

Now:

*'s(ak) Along-track distance of sample **ak.**'*

3. Section 5.3.3.4, step 5.3.3.4.1, p69

Typo corrected. Text was:

'The sequence of times t(k) is defined by

$$a. t(ak) = T_0 + (ak - K)\Delta T, ak = 0, 1, 2, 3, \dots, '$$

Now:

*'The sequence of times t(**ak**) is defined by
t(ak) = T₀ + (ak - K)ΔT, ak = 0, 1, 2, 3, ..., '*

4. Section 5.3.3.4, step 5.3.3.4.1, p70



Typo corrected. Text was:

'The required outputs are the latitude and longitude of the subsatellite point at $t(k)$.'

Now:

'The required outputs are the latitude and longitude of the subsatellite point at $t(a_k)$.'

5. Section 5.3.3.4, step 5.3.3.4.1, p71

Typo corrected. Text was:

'The length of the line segment $ds[k]$ between points $k - 1$ and k is then derived by a call to the common function normal section length [and azimuth].'

Now:

'The length of the line segment $ds[a_k]$ between points $a_k - 1$ and a_k is then derived by a call to the common function normal section length [and azimuth].'

6. Section 5.3.3.4, step 5.3.3.4.1, p71

Typo corrected. Text was:

'The required output is the line element $ds(k)$.'

Now:

'The required output is the line element $ds(a_k)$.'

7. Section 5.3.3.4, step 5.3.3.4.1, p71

Typo corrected. Text was:

' $s(a_k) = s(a_{k-1}) + ds(a_k)$, $k = 1, 2, \dots$ eq 5.3-25'

Now:

' $s(a_k) = s(a_{k-1}) + ds(a_k)$, $a_k = 1, 2, \dots$ eq 5.3-25

Re-define origin of y

For all a_k

$track_y(a_k) = s(a_k) - s(K)$

,

8. Section 5.3.3.4, step 5.3.3.4.1-2, p71

Email comment (8) from IM, 31/08/10



I do not understand why the loop over the Tie points index i is between $ak-K$ and $ak=K$ (I guess that the '=' should be replaced by a '+'). For me, this loop is useless and the loop over each ak , mentioned just above step 5.3.3.4.2 is sufficient, provided that the ak index replaces every i index. **Do you agree ?**

Text was:

'The following steps are then implemented for all ak :

Step 5.3.3.4.2 Generate Geolocation Grids

Starting with the tabular along-track latitudes and longitudes, the latitude and longitude of every across-track tie point corresponding to the along track grid points are calculated, using spherical trigonometry. There are $N1 + N2 + 1$ tie points spaced by dL km. The following calculations are repeated for each value of $i = ak-K, ak=K$.'

Now:

'Step 5.3.3.4.2 Generate Geolocation Grids

Starting with the tabular along-track latitudes and longitudes, the latitude and longitude of every across-track tie point corresponding to the along track grid points are calculated, using spherical trigonometry. There are $N1 + N2 + 1$ tie points spaced by dL km. The following calculations are repeated for each value of $i = ak-K$, where $0 < i < \text{number of tie points in an orbit}$ '

9. Section 5.3.3.4, step 5.3.3.4.2.1, p71

Typo. Text was:

'grid_long($i, N1$)'

Now:

'grid_long(i, N_1)'

10. Section 5.3.3.4, step 5.3.3.4.2.4, p72

Email comment (8) from IM, 31/08/10

Is there any reason for doing this after the computation of the lat/lon of every Tie Points ? eq 5.3-40 could be applied earlier just after eq 5.3-25

Delete the following from this location (has been put in elsewhere)

'Step 5.3.3.4.2.4 Redefine origin of y .

For all k

$$track_y(ak) = s(ak) - s(K) \text{ eq 5.3-40'}$$



11. Section 5.3.3.4, step 5.3.3.4.2.4, p72

Email comment (8) from IM, 31/08/10

The x coordinate of the Tie Points grid should be computed as well according to SY-4. I included it while computing the lat/lon of the Tie Points grid in the overall step 5.3.3.4.2., do you agree ?

x track =0 on the ground track. NO CHANGE MADE.

12. Section 5.3.3.6, step 5.3.3.4.2.4, p79

Typo. Text was:

status = xp_target_extra_target_to_sun(parsmeters)

Now:

status = xp_target_extra_target_to_sun(parameters)

13. Section 5.3.3.6, p74

Email comment (2) from IM 31/08/10

section 4.4.4.4. and eq 5.3-46: it seems as if the platform reference frame should correspond to the reference frame used by the CFI, since matrix M_{ps} defined in eq 5.3-45 is supposed to represent the transformation from the satellite reference frame (the one defined by TAS-F) to the one used in the CFI. However, the satellite reference frame defined in the CFI does not correspond to your platform reference frame (see section 5.2 of the MCD, issue 4.1, EO-MA-DMS-GS-0001), could you please check this and confirm if there is a mistake ?

Eqn 5.3-46 was:

$$M_{ps} = \begin{vmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Now:

$$M_{ps} = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{vmatrix}$$



14. Section 5.4.3.1.7, p 97

Email comment (4) from IM, 01/09/10

* //Several parameters are not explained in the SY-4 document or in the ATBD document/
/Expected_scan_value,
Pos_Err,
Expected_Flip_value,
Av_Position_Err
and the different thresholds.

Expected_Flip_Value not consistent with the name in SY-4 and should only be indexed by view.
Text was:

```
'for(i=0, i<TargetLength[iview], i++){  
    InstantaneousError[i] = FlipEncoderValue[i,iview] - ExpectedFlipValue[i,iview];'
```

Now:

```
'for(i=0, i<TargetLength[iview], i++){  
    InstantaneousError[i] = FlipEncoderValue[i,iview] - Flip_nominal_position[iview];'
```

Av_position_Err is calculated within the ATBD. The equations are provided in the text.
NO CHANGE TO DOCUMENT

15. Section 5.5.3.3, step 5.5.3.4, p128

Email comment (4) from IM, 31/08/10

Text was:

'Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled on the Tie Point grid for the product.'

Now:

'Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled *using bilinear interpolation in x, y from the full resolution grid* on the Tie Point grid for the product.'

Cloud queries:

16. Section 5.5.3.6.1, p134

Comment from CH?

A/ In the _spatial coherence test_ :

1°) I don't understand how pixel positions of the groups are calculated:



In section 5.5.3.6.1, they are calculated with respect to the satellite ground track corresponding to pixel number M where $x=0$ and their pixels position are given by $[(x_index*3) - M: (x_index*3)- M+2 , y_index*3 : y_index*3+2]$.

However:

- * According to my understanding, M is an across-track index and should be taken into account in the calculation of y_index .
- * if $M > 3$, pixel positions can be negative
- * and finally, this definition implies that only one side of the image is taken into account

The constant M is now clarified in the text and a diagram inserted to make the coordinate system clear.

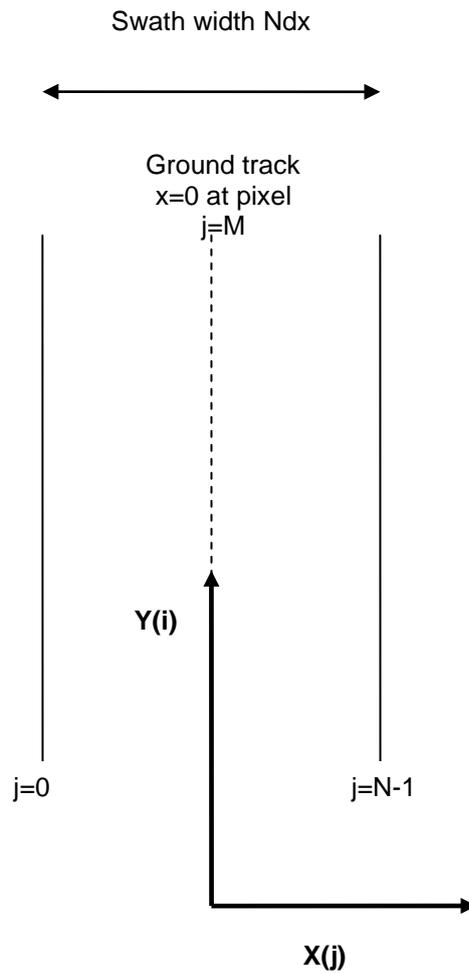
Text was:

'The test is performed separately for the nadir and oblique views. The nadir view 12 μ m and 11 μ m BT data are used in the nadir view test and the oblique view 12 μ m and 11 μ m BT data are used in the oblique view test. In the following, it is assumed that the size of the nadir and oblique swaths are N km, where N will differ for each viewing direction.'



Now text is as below, and with diagram for clarification of coordinate system:

The test is performed separately for the nadir and oblique views. The nadir view 12 μ m and 11 μ m BT data are used in the nadir view test and the oblique view 12 μ m and 11 μ m BT data are used in the oblique view test. In the following, it is assumed that the size of the nadir and oblique swaths are N km, where N will differ for each viewing direction. The across track co-ordinate is x and is indexed by j, the along track co-ordinate is y and is indexed by i. In the equations that follow, the constant M is the pixel number at x=0, the ground track. The figure below shows the coordinate used in this test.



NB nadir and oblique swaths have different dimensions so M and N are view dependant



17. Section 5.5.3.6.1, step 5.5.3.6.1-8, p137

Replace duplicate use of i and j indices

Text was:

*'where the summations are for indices in the ranges of
 $x_index-3 < i < x_index + 3$ and
 $y_index-3 < j < y_index + 3$
i and j must also be in the range of'*

Now;

*'where the summations are for indices in the ranges of
 $x_index-3 < u < x_index + 3$ and
 $y_index-3 < v < y_index + 3$
u and v must also be in the range of'*

18. Section 5.5.3.6.1, P134

Comment from CH?

2°) Could you confirm that the dimension of the product grid in the along track direction depends on the view (nadir or oblique) but not the dimension in the across track direction

No – the dimension of the product grid in the across-track direction will depend on the view, because it is essentially proportional to the width of the swath, which is narrower for the oblique view than for the nadir view. The dimension in the along-track direction will be the same for both views.

NO CHANGE MADE TO DOCUMENT.

19. Section 5.5.3.6.1, step 5.5.3.6.1, p134

3°) In step 5.5.3.6.1, Does the test take into account COH_AREA_SIZE_X or COH_AREA_SIZE_Y ?

The large-scale component of the spatial coherence tests (Steps 5.5.3.6.1-9 onwards) uses COH_AREA_SIZE_X and COH_AREA_SIZE_Y to define the dimensions of the rectangular sub-areas for which dynamic thresholds are calculated.

NO CHANGE MADE TO DOCUMENT.

(IR histogram tests to be checked again by ARB- these changes may alter so do not put into doc yet)

20. Section 5.5.3.6.9, step e1, p156



Comment from CH?

2°) Some parameters of the IR histogram test seem to be unused (no quotation in the ATBD): `peak_frac_min`, `second_low_fraction` and `ir_spread_na/ob`

Text was:

'Starting at the histogram peak and working to the left, inspect each histogram bin until either a bin value of zero or a local minimum is found.'

Now:

*'Starting at the histogram peak and working to the left, inspect each histogram bin until either a bin value of zero or a local minimum **lower than `peak_frac_min` times the peak value** is found.'*

21. Section 5.5.3.6.9, step i.2.2, p157

Comment from CH?

2°) Some parameters of the IR histogram test seem to be unused (no quotation in the ATBD): `peak_frac_min`, `second_low_fraction` and `ir_spread_na/ob`

Text was:

'- the value of the major peak at its higher limit is greater than half the minor peak'

Now:

*'- the value of the major peak at its higher limit is greater than **`second_low_fraction` times** the minor peak'*

22. Section 5.5.3.6.9, step j, p158

Comment from CH?

2°) Some parameters of the IR histogram test seem to be unused (no quotation in the ATBD): `peak_frac_min`, `second_low_fraction` and `ir_spread_na/ob`

Add another step. Text was:

'j) Choose between the lower limits of the major and the minor peaks, if necessary:

*If both peaks are VALID then choose the lower value.
If there is only one VALID peak then choose its lower limit.
If there is no VALID peak then flag all pixels as cloudy and exit.'*

Now:

'j) Choose between the lower limits of the major and the minor peaks so that that if necessary:



*If both peaks are VALID then choose the lower value.
If there is only one VALID peak then choose its lower limit.
If there is no VALID peak then flag all pixels as cloudy and exit.*

j.1) For the chosen peak, calculate the difference

*(peak_interval[CHOSEN] – low_limit[CHOSEN])*bin_size*

(where bin_size = dT = 0.1 K). If this quantity is greater than the value of ir_spread appropriate to the view, then replace the previous threshold value by

threshold = peak_interval[CHOSEN] – ir_spread[view]/bin_size

and omit step (k), otherwise execute step (k).'

New changes in release 3 of change sheet – 29-Oct-2010

23. Section 5.4.3.4, p105

Typo to be corrected.

Text was:

'time_offset is the period between the time when the VISCAL is fully illuminated and the time that Sentinel-3 crosses the terminator, ~ 4.8 TBD minutes. (The time of full illumination precedes the terminator crossing.)'

Now

*'time_offset is the period between the time when the VISCAL is fully illuminated and the time that Sentinel-3 crosses the terminator, ~ 4.8 TBD minutes. (The time of full illumination **occurs after** the terminator crossing.)'*

24. Section 5.5.3.6, p133 Cloud tests

Email comment 24/09/10

1. VERY IMPORTANT POINT: the handling of the different product grids seems to not be taken into account, especially in the snow covered surface tests, and in the 1.6 and 2.25 histogram tests that uses data from different channels that are given on different grids. Could you please provide detailed clarifications on all the points of the cloud identification algorithm concerned by this specific problem, and particularly explain how data shall be retrieved from one grid to another one (interpolation ?, other method) ?



In the cloud section introduction, text is added to clarify this point. After the second paragraph (beginning 'the 1.6 micron test.....') and before the third paragraph (beginning 'In general....') on p134, add the following text:

'The cloud tests are performed at the native resolution of the detector elements – i.e. for TIR detectors cloud flags are computed at 1km resolution and VIS/SWIR at 0.5km resolution. In detail, the cloud tests are all based on the regridded brightness temperatures or reflectances, which are sampled on a rectangular grid in the cartesian co-ordinates X and Y, where X is the across-track co-ordinate and Y is the along track co-ordinate. For each view there are two grids, at 1 km and 0.5 km resolution, as appropriate, so the grid for the solar channels has twice the dimensions of that for the solar thermal channels. The 1.0 and 0.5 km image grids are coincident in the sense that alternate samples of the 0.5 km grid coincide with the samples of the 1.0 km grid.

We adopt the convention that the grid is indexed by i, j , where the index i is an along-track index (to the rows of the image array) and j is the across-track index (to the columns). With this convention, suppose the thermal image arrays have N columns, indexed $j_{.05} = 0$ to $N - 1$. Then the solar channel arrays will have $2N$ columns indexed $j_{.05} = 0, 2N-1$. Similarly we can index the arrays in the along-track direction with $i_{.10}$ and $i_{.05}$ for the 1.0 km and 0.5 km channels respectively.

In those cases where a test based on the solar channels uses thermal channel data, the thermal channel value associated with the solar channel pixel at $[j_{.05}, j_{.05}]$ is that indexed by

$$i_{.10} = \text{integer part of } (i_{.05}/2)$$

$$j_{.10} = \text{integer part of } (j_{.05}/2)$$

*(In other words the geometrical instrument pixel $[i_{.10}, j_{.10}]$ at 1 km resolution includes the four 0.5 km pixels $[2*i_{.10}, 2*j_{.10}]$, $[2*i_{.10}+1, 2*j_{.10}]$, $[2*i_{.10}, 2*j_{.10}+1]$, $[2*i_{.10}+1, 2*j_{.10}+1]$). The 0.5km grid replicates the 1km cloud tests for each pixel.*

In any case where a test based on the thermal channels uses solar channel data, the Logical OR of the 0.5km flags within that 1km pixel are used. Note there is no interpolation.'

25. Section 5.5.3.6.3, p142

Email comment 24/09/10

3. Could you please provide the exact definition of the LUT to be included in an ADF to extract the AC zone number used in the thin cirrus and fog/low stratus tests ?

Tex was:

'a) Determine the pixel across track band number (band_no) using the across-track co-ordinate of the pixel for x , and the band coordinates provided in the cloud LUT.'

Text now:

'a) Determine the pixel across track band number (band_no) using the across-track pixel number and the band table provided in the cloud LUT.



across-track band for pixel $j = \text{band_table}[\text{abs}(j)]$

26. Section 5.5.3.6.3, p143

Email comment 24/09/10

3. Could you please provide the exact definition of the LUT to be included in an ADF to extract the AC zone number used in the thin cirrus and fog/low stratus tests ?

Text was:

'b) Determine the across track band number band_no from the pixel x coordinate.using equation 5.6.1.'

Now:

'b) Determine the pixel across track band number (band_no) using the across-track pixel number and the band table provided in the cloud LUT.

across-track band for pixel $j = \text{band_table}[\text{abs}(j)]$

27. section 5.5.3.6.1, p135

3°) In step 5.5.3.6.1, Does the test take into account COH_AREA_SIZE_X or COH_AREA_SIZE_Y?

In addition, we have to calculate the appropriate COH_MIN_DIF according to the nadir swath zones. Which pixel is taken into account to determine Same question for COH_AREA_DIF and COH_AREA_THRESH.

Text was:

'The sub areas are, similarly to the 3x3 groups above, non-overlapping and congruent.

For each group do Steps 5.5.3.6.1-1 to 5.5.3.6.1-5 inclusive:'

Text now:

'The sub areas are, similarly to the 3x3 groups above, non-overlapping and congruent.

To determine which of the nadir zones the sub-areas fall into in the case of a sub-area not falling entirely into one of the zones defined by $|X1|$ or $|X2|$, we use the parameters of the zone with the greatest overlap. For example, if a zone defined by $Xa: Xb$, where $Xa < X1 < Xb$

Use the zone $-X1$ to $X1$ if $(X1-Xa) < (Xb-X1)$

Use the zone $|X1|$ to $|X2|$ otherwise

For each group do Steps 5.5.3.6.1-1 to 5.5.3.6.1-5 inclusive:'



28. Section 5.3.3.4.2, p71

- o The x coordinate of the Tie Points grid should be computed as well according to SY-4. I included it while computing the lat/lon of the Tie Points grid in the overall step 5.3.3.4.2., do you agree ?

Text was:

'Starting with the tabular along-track latitudes and longitudes, the latitude and longitude of every across-track tie point corresponding to the along track grid points are calculated, using spherical trigonometry. There are $N_1 + N_2 + 1$ tie points spaced by dL km. The following calculations are repeated for each value of $i = ak-K, ak=K$.'

Now:

*'Starting with the tabular along-track latitudes and longitudes, the latitude, longitude **and x-coordinate** of every across-track tie point corresponding to the along track grid points are calculated, using spherical trigonometry. There are $N_1 + N_2 + 1$ tie points spaced by dL km. The following calculations are repeated for each value of $i = ak-K, ak=K$.'*

29. Section 5.3.3.6, p74

- * step 5.3.3.6.1: I guess that the scan angle ϕ_p can be deduced from the positions of the scan mirror given in the scan position packet. However, I did not see a method in the source packet processing enabling to convert them into an angle directly useable in the geolocation process. Could you please provide the algorithm to be used ?

Text was:

'Let ϕ_p be the scan angle of the pixel, taken from the spacecraft telemetry.'

Now:

*'Let ϕ_p be the scan angle of the pixel. **The scan mirror encoder position taken from the spacecraft telemetry is converted into ϕ_p using the encoder_conversion_nadir and encoder_conversion_oblique parameters in the geometry ADF.**'*

30. Section 5.4.3.1.7, p92

- * The "general limit checks" is applied to all converted items for



which surveillance limits are specified. What is the default value of surveillance limits for item not concerned by this “general limit checks”?

text was:

‘This test is applied to all converted items for which maximum and minimum surveillance limits are specified in the ancillary data ADF.’

Now:

‘This test is applied to all converted items for which maximum and minimum surveillance limits are specified in the ancillary data ADF. If no limit is specified, then no check is made.’

31. Section 5.1.2.1.1 p50

Few quality flags have been defined in SLSTR Level 0 product as annotations (see table 6-11 of SLSTR Product Definition SY-4). However, the algorithm enabling to compute them at Level 0 is not defined in the ATBD. Could you please provide it as soon as possible, so that we can consolidate our specifications on Level 0 processing in SLSTR DPM ?

Text was:

‘The GPP perimeter document [RD-10] does not require quality checking, on the assumption that the input stream is perfect, but possible additional quality tests might include:

- 1. CRC check,*
- 2. Consistency of the scan count with the time stamp,*
- 3. Valid fixed header information,*
- 4. Valid PCAT codes,*
- 5. Valid target ID codes,*
- 6. Consistent sequence counter values,*
- 7. Duplicate packets. ‘*

Text now:

The GPP perimeter document [RD-10] does not require quality checking, on the assumption that the input stream is perfect, but the following ISP checks are performed:

Set all flags for the ISP to 0 (i.e. all OK)

- 1. CRC check*

The CRC check will follow the scheme as defined in Annex A of ECSS--E--70--41A

If error is present then

CRC Error flag for ISP(i) = 1b

- 2. Consistency of the scan count with the time stamp,*

The test to be performed checks that the time interval between successive scans is nominally 301ms as follows.



Note: Care has to be taken to account for the wrap around of the packet sequence counters and instrument scan counters that should occur every 2^{16} scans. If a negative value is detected add 2^{16} to the value before comparing.

$$\Delta t = (\text{time}(\text{scan}_n) - \text{time}(\text{scan}_{n-1})) / (\text{scan_count}(\text{scan}_n) - \text{scan_count}(\text{scan}_{n-1}))$$

IF $\Delta t \neq 301 \pm dt$ then

scan_count_discontinuity flag for scan_n = 1b

Where dt is the resolution of the on board clock

3. *Valid fixed header information*

If (fixed_header_value(ISP(i)) \neq expected_fixed_header_values) then

Header_Error(ISP(i)) = 1b

4. *Valid PCAT codes,*

This test checks that the Packet IDs contained in the ISPs fall within the expected range. Values outside this range could indicate a non standard instrument configuration.

If (PCAT (ISP(i)) > Chex) then

Invalid_PCAT(ISP(i)) = 1b

5. *Valid target ID codes*

This test checks that the target IDs contained in the ISPs fall within the expected range. According to RD-6, the baseline set of target ID's falls in the range A0h to D1h. Values outside this range could indicate a non standard instrument configuration.

If (Target_ID(ISP(i)) lies outside of range 0Fh to D2h) then

Invalid_Target_ISP(ISP(i)) = 1b

6. *Consistent sequence counter values*

This test checks that the packet counter is increasing monotonically. A difference between successive packets not equal to 1 could indicate a missing packet or data out of sequence.

Note: As with step 2, care are has to be taken to account for the wrap around of the packet sequence counters and instrument scan counters that should occur every 2^{14} scans. If a negative value is detected add 2^{14} to the value before comparing.



If($seq_count(isp(i)) - seq_count(isp(i-1)) \neq 1$) then
Sequence_Error($isp(i)$) = 1b

7. Duplicate packets.

This test checks that the packet is not a duplicate of one previously processed. The test compares the sequence_count, scan_count and crc_error, time of the ISP against the previous value. If all fields are identical then the packet shall be flagged as duplicate and is not to be used in the processing.

Duplicate_Packet($isp(i)$) = 1b'

32. Section 5.4.31.

Wrong section number reference

Text was:

'Section 3.1.1 has described the unpacking of the Level 0 product'

Now

'Section 5.1 has described the unpacking of the Level 0 product'

33. Section 5.4.3.1.1 p 86

Wrong section number reference and array name reference incorrect

Text was:

'In the present formulation, these tests have been performed during the stage of data input and unpacking described in Section 3.1.1, and their results can be found in the array of flag structures $ISP_test[i][j]$ described there.

A single overall validity flag for the ISP of type j corresponding to scan i can be defined as the union (logical OR) of the individual flags:

$$ISP_invalid[i][j] = \bigcup_k ISP_test[i][j][k]'$$

Now:

'In the present formulation, these tests have been performed during the stage of data input and unpacking described in Section 5.1.2 and their results can be found in the array of flag structures $quality[i][j]$ described there.

A single overall validity flag for the ISP of type j corresponding to scan i can be defined as the union (logical OR) of the individual flags:

$$ISP_invalid[i][j] = \bigcup_k quality[i][j][k]'$$



Changes added in version 4 of change sheet – 03-Nov-2010

34. Section 5.5.3.2, p120

Typo corrected. Text was:

$$R_C(S, p'_C, j) = R_B(S, p'_C - 1, j) \text{ if } p'_C = 2n_t - 1$$

Text now:

$$R_C(S, p'_C, j) = R_B(S, p'_C - 1, j) \text{ if } p'_c = 2n_t$$

Changes added in version 5 of change sheet – 10-Nov-2010

35. Section 5.4.3.4, p10

Email comment Tue 02/11/2010

section 5.4.3.4: the channels that should be denoted by ch should be S1-S6 and not S6-S9 F1-F2, do you confirm ?

Text was:

'Where ch denotes the channel (S6-S9, F1-2),'

Text now:

Where ch denotes the channel (S1-S6),'

36. Section 5.4.3.3, p101

Email comment Tue 02/11/201

step 6: I assume that the radiometric noise should be computed on valid pixel only, do you confirm ? If yes, could you please add this detail in the ATBD.

Clarify that all operations are performed on valid pixels

Text was:

'The following five steps are applied to all packets in a calibration period.'

Text now:

'The following five steps are applied to all packets in a calibration period for all valid pixels.'

37. Section 5.4.3.4, p105

Email comment Tue 02/11/201



general introduction: Regarding the handling of parity for solar channels, I am not sure any more that I did well in the DPM after reading the new version of ATBD that includes this detail. I assumed that the parity is the same as the one of t. Do you confirm ?

text was:

'and t defines the acquisition cycle.'

Text now:

'and t defines the acquisition cycle (which is equivalent to parity).'

38. Section 5.4.3.4, p107

Email comment Tue 02/11/201

Do you confirm that reference channel for detection (i.e. 0.87) is the channel called detection_channel in the processing control parameter ADF defined in SY-4 ?

On the other hand, there is a specific detector element (identified as detection_element in SY-4) to be used to do the detection, but it is not clearly identified in the text of the ATBD. Could you please clarify ?

Do you confirm that the monitor_threshold of the ATBD corresponds to the processing control parameter detection_threshold defined in SY-4 ?

Text was:

'By default, the monitoring channel will be the 0.87 μ m channel, although there should be flexibility for this to be altered.'

Define monitor_count[view; s] as average counts from 0.87 (or otherwise) channel averaged over all pixels and detector elements(?) as a function of scan cycle s.

Here, we find the period of time when the monitor channel counts are above a certain threshold value (monitor_threshold) to ensure the VISCAL is in full solar illumination.'

Text now:

'By default, the monitoring channel will be the 0.87 μ m channel, although there should be flexibility for this to be altered and this is allowed for by including the parameter detection_channel in the SLSTR specific auxiliary data file. The detector element to use for monitoring is detection_element in the SLSTR specific auxiliary data file.'

Define monitor_count[view; s] as average counts from 0.87 (or otherwise) channel averaged over all pixels as a function of scan cycle s.

Here, we find the period of time when the monitor channel counts are above a certain threshold value (monitor_threshold) to ensure the VISCAL is in full solar illumination. The monitor_threshold is equivalent to the detection_threshold in the auxiliary data file'



35. Section 5.4.3.4, p107

Email comment Tue 02/11/201

the threshold corresponding to the minimum number of scans necessary to derive valid calibration coefficients is not part of the processing control parameters defined in SY-4. To me it is indispensable that it is parametrizable. Could you please add this threshold to the processing control parameters defined in SY-4.

detection_scan_threshold to be added to ADF. Text was:

'If there are fewer than 17 (now 17*2?)TBD scans present in the period between the start of the VISICAL window and the end of the Level 0 data (this situation might arise for NRT data) an initial value of the smoothed monitor count, mon1, cannot be derived. In this case a valid VISICAL product cannot be generated, and solar calibration parameter processing should be abandoned.

If last_s - first_s ,17 TBD then abandon'

Text now:

'If there are fewer than **detection_scan_threshold** scans present in the period between the start of the VISICAL window and the end of the Level 0 data (this situation might arise for NRT data) an initial value of the smoothed monitor count, mon1, cannot be derived. In this case a valid VISICAL product cannot be generated, and solar calibration parameter processing should be abandoned.

If last_s - first_s < **detection_scan_threshold** then abandon'

35. Section 5.4.3.4, p107

Email comment Tue 02/11/201

Could you please explain the process of 'smoothing the monitor_count'? I do neither understand the formula nor the text that accompanies it.

Text was:

'Let the smoothing interval be sm_int=8 TBD now I am using scan cycle rather than scan, this is prob 5
Start with

scan s = (first_s + sm_int)

Calculate a smoothed monitor count mon1 for the first sample:

mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0

If mon1 > monitor_threshold, it may not be possible to derive a valid centroid, and processing should be abandoned.

Otherwise do the following while statements so that we find the value of s when mon1>monitor_threshold:

while mon1 < monitor_threshold and s < (last_s - sm_int)



```
s = s + 1
mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0
end while'
```

Text now:

'Let the smoothing interval be sm_int

```
sm_int = 1
```

The index of the first scan to be processed is not first_s but s0, where

```
s0 = (first_s + sm_int)
```

Calculate a smoothed monitor count mon1 for the first sample where the centre of the interval is given more weighting:

```
mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0
```

where s is greater than or equal to s0

If mon1 > monitor_threshold, it may not be possible to derive a valid centroid, and processing should be abandoned.

Otherwise do the following while statements so that we find the value of s when mon1>monitor_threshold:

```
while mon1 < monitor_threshold and s0 < s < (last_s - sm_int)
s = s + 1
mon1 = (monitor_count[s-sm_int] + 2 * monitor_count[s] +
monitor_count[s+sm_int])/4.0
end while'
```

35. Section 5.4.3.4, p107

Email comment Tue 02/11/201

When talking about source_packet_ut_time, do you refer to the current UTC scan time ?

Text was:

' When the while loop ends, it defines the first value of s of the scan cycles for which the smoothed monitor count exceeds the threshold. '

Text now:

'When the while loop ends, it defines the first value of s of the scan cycles for which the smoothed monitor count exceeds the threshold. *We also find the source_packet_ut_time, the UTC time of packet s.*'

36. Section 5.4.3.1.7, p96



Email comment (4) from IM, 01/09/10

```
* / /Several parameters are not explained in the SY-4 document or in
the ATBD document/:
/Expected_scan_value,
Pos_Err,
Expected_Flip_value,
Av_Position_Err
and the different thresholds.
```

Text was

```
`SumsqPositionError = 0;
NumSamples = 0;
for(i=0, i<TargetLength[iview], i++){
InstantaneousError[i] = ScanEncoderValue[i,iview] - ExpectedScanValue[i,iview];`
```

text now:

```
`SumsqPositionError = 0;
NumSamples = 0;
for(i=0, i<TargetLength[iview], i++){
ExpectedScanValue(i) = target first acquisition + i
InstantaneousError[i] = ScanEncoderValue[i,iview] - ExpectedScanValue[i,iview];`
```

37. Section 5.4.3.4, p105

step 1: what do you call terminator ?

Text was:

'time_offset is the period between the time when the VISCAL is fully illuminated and the time that Sentinel-3 crosses the terminator, ~ 4.8 TBD minutes. (The time of full illumination precedes the terminator crossing.)'

Text now:

'time_offset is the period between the time when the VISCAL is fully illuminated and the time that Sentinel-3 crosses the terminator, ~ 4.8 TBD minutes. (The time of full illumination precedes the terminator (day-night boundary) crossing.)'

38. Section 5.4.3.4, step7, p111

List of variables (from SY-4) for which the corresponding computation algorithm might be missing in the ATBD: L_VISCAL

Text was:

'and where the values of mean_solar_irradiance are taken from the ancillary file of configuration data.'

Text now:



'and where the values of *mean_solar_irradiance* are taken from the ancillary file of configuration data. We can also calculate the *VISCAL* radiance, *L_viscal*

$$L_viscal(ch, view, k) = Reflectance\ factor(ch, view, k) \times solar\ irradiance(ch, view, k) / \pi$$

Changes Added in Version 6 – 11 November 2010

39. Section 5.4.3.4, step7, p111

section 6.8.3.4: dimension of the solar irradiance depends on the bands and on the detectors. However, I do not see the dependency on detectors in the ATBD, and I do not really understand why solar irradiance should depend on the detector. Could you please clarify and confirm if a detector dependency should be taken into account or not ?

Text was:

$$solar_irradiance[ch] = mean_solar_irradiance[ch]^*$$

Text Now:

$$solar_irradiance[ch, view, k] = mean_solar_irradiance[ch, view, k]^*$$

Changes Added in Version 7 – 10 December 2010

40. Section 5.5.3.3, p121

All comments on the regridding process addressed.

i) Text on p121 was

The grid of image points is based on sampling intervals of 1 km and 0.5 km for the thermal and solar channels respectively. In the along-track direction, following the AATSR convention, equal time interval sampling is used. In this scheme the interval Δy between successive image rows is $v\Delta t$, where V is the ground trace velocity of the sub-satellite point, and Δt is a fixed fraction of the scan period, chosen so that the number of image rows in a given interval equals the number of instrument scans. Thus if τ is the scan period, $\Delta t = 0.5\tau$ for the thermal channels, and $\Delta t = 0.25\tau$ for the solar channels. Consequently the sampling interval in the along-track direction is approximately equal to that in the across-track direction, but varies around the orbit in proportion to the ground trace velocity.

The grid is characterised by indices i and j in the along-track and across-track directions respectively, such that the x and y co-ordinates of the image point indexed by (i, j) are

$$\begin{aligned} x(i, j) &= x_0 + j\Delta x \\ y(i, j) &= y_0 + (i - i_0)\Delta y \end{aligned} \qquad \text{eq 5.5-1}$$

In these equations x_0 is a constant representing the x co-ordinate of the image point at $j = 0$, and y_0 is a local origin in y corresponding to image row i_0 .



The x and y co-ordinates of each instrument pixel have been calculated using the algorithms in sections 3.2.3.6, 3.2.3.7. If the co-ordinates of a given instrument pixel P are x_p, y_p , then the core algorithm for remapping the pixel is

$$i = i_0 + \text{integer part of } [(y_p - y_0) / \Delta y]$$

$$j = \text{integer part of } [(x_p - x_0) / \Delta x]$$

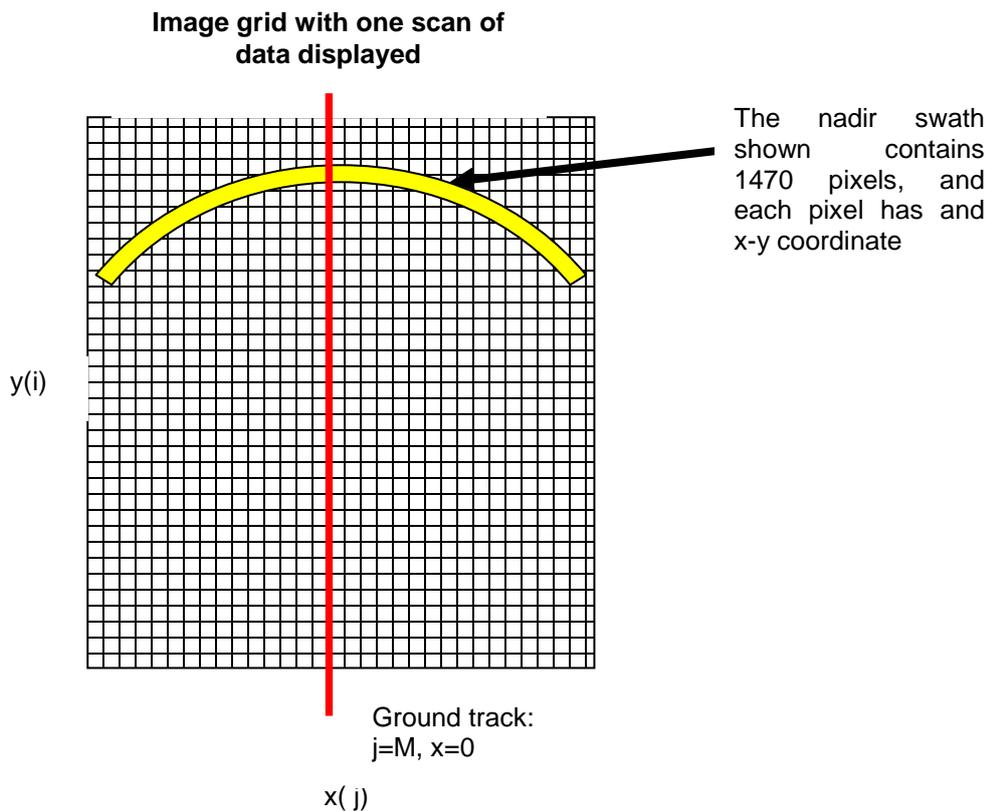
eq 5.5-2

The pixel P is mapped into the image array position (i, j) . (Note that following the AATSR convention the integer part is used, not the nearest integer.)

Text now:

There are two image grids onto which the measurement pixels should be regridded: a 1km grid and a 0.5km grid. Each channel of each view must be regridded separately. Here we outline the approach that is adopted for a single combination of channel and view.

The starting point for a given channel/view is a set of brightness temperature or reflectance values, sampled on an instrument grid which reflects the curved scanning geometry of the SLSTR instrument, together with the X and Y co-ordinates of these instrument pixels in the image co-ordinate system.



The aim of the regridding algorithm is to resample this set of instrument pixels onto a uniform rectangular grid of points in X and Y . Here X is the across-track co-ordinate and Y is the along track co-ordinate.



We adopt the convention that the image grid is indexed by i, j , where the index i is an along-track index (to the rows of the image array) and j is the across-track index (to the columns). This can seem confusing, because i maps to Y and j to X , but it is consistent with the usual convention for indexing arrays, where $A[i,j]$ represents an array indexed by rows (i) and columns (j).

The image is assumed to comprise N columns, indexed $j = 0$ to $N - 1$. The columns are equally sampled in X at interval dX , so that there is a linear relationship between X and j , which may be written as

$$(1) X(j) = (j - M)dX.$$

where M is a constant such that the column indexed by $j = M$ has $X = 0$ and so corresponds to the nominal ground track, since the definition of X states that $X = 0$ on the ground track. The nominal swath width is NdX . For the thermal channels, $dX = 1$ km, for the solar channels $dX = 0.5$ km.

In the along-track dimension the situation is more complicated, because, following AATSR practice, equal time sampling has been adopted. This means that the sampling interval in Y is given by

$$(2) dY = v dT,$$

where v is the ground track velocity of the satellite, and dT is the appropriate fraction of the scan (SCANSYNC) period. That is, $dT = \text{SCANSYNC} / K$, where K is the number of detector rows for the channel. The ground track velocity of the satellite varies around the orbit, so the spatial sampling interval dY also varies and we cannot assume a linear relationship between Y and the index i . Locally however the relationship is linear to a close approximation. If $Y(i)$ is the Y co-ordinate of the row of samples indexed by i , we can write

$$(3) Y(i + k) = Y(i) + kdY$$

where dY is locally constant, and is approximately equal to dX .

The set of points $[i, j]$ whose co-ordinates are $[X(j), Y(i)]$ for all valid i and j defines the sampling grid. Consider the rectangle whose corners are at the four points $[X(j), Y(i)], [X(j), Y(i+1)], [X(j+1), Y(i+1)], [X(j+1), Y(i)]$; this is a small rectangular area of approximate side 1 km (thermal channels) or 0.5 km (solar channels), and can be taken to define the image pixel at grid point $[j, i]$. The centre of this area is at $[X(j) + 0.5dX, Y(i) + 0.5dY]$, and the set of points $[X(j) + 0.5dX, Y(i) + 0.5dY]$ form the grid of pixel centres, displaced relative to the grid of corner points $[X(j), Y(i)]$ by the vector $[0.5dX, 0.5dY]$.

Relationship between the image grids

As noted above, there are two image grids, at nominal sampling intervals of 1.0 and 0.5 km for the thermal / fire and solar channels respectively. In order to distinguish between the two, let us adopt the notation that the 1.0 km grid is indexed by $[i_{10}, j_{10}]$, and the 0.5 km grid by $[i_{05}, j_{05}]$. Note that as before i is the along-track and j is the across-track index.

The 1.0 km image grid therefore consists of the set of points having co-ordinates

$$X[i_{10}, j_{10}], Y[i_{10}, j_{10}]$$

where

$$X[i_{10}, j_{10}] = (j_{10} - M)dX$$

and

$$Y[i_{10} + 1, j_{10}] = Y[i_{10}, j_{10}] + dY$$

The 0.5 km grid consists of the set of points having co-ordinates

$$X[i_{05}, j_{05}], Y[i_{05}, j_{05}]$$

where

$$X[i_{05}, j_{05}] = (j_{05} - M)(dX / 2)$$



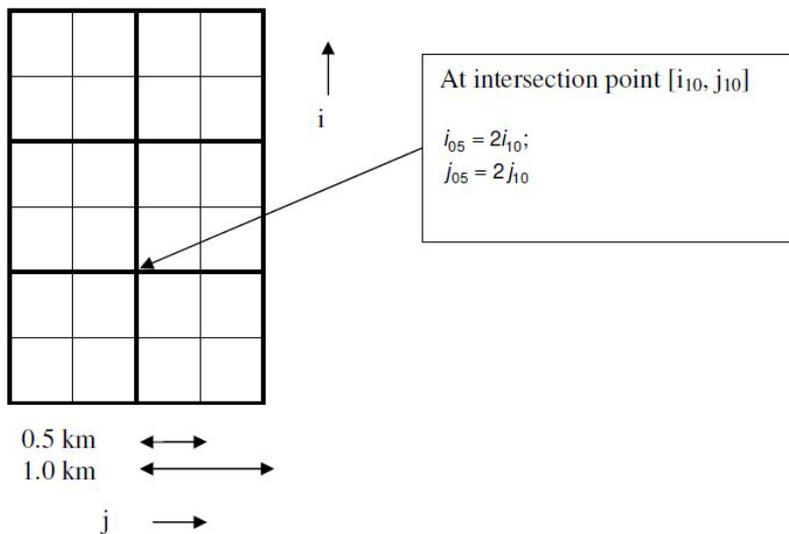
and

$$Y[i_{05} = 2i_{10}, j_{05}] = Y[i_{10}, j_{10}]$$

In other words the points of the 1.0 km image grid having the indices $[i_{10}, j_{10}]$ are coincident with the points of the 0.5 km grid having indices

$$i_{05} = 2i_{10}; j_{05} = 2j_{10}$$

We regard the grid points as defining the lower left corners (not the centres) of the image pixels. (In other words the geometrical instrument pixel $[i_{10}, j_{10}]$ at 1 km resolution includes the four 0.5 km pixels $[2 \cdot i_{10}, 2 \cdot j_{10}]$, $[2 \cdot i_{10} + 1, 2 \cdot j_{10}]$, $[2 \cdot i_{10}, 2 \cdot j_{10} + 1]$, $[2 \cdot i_{10} + 1, 2 \cdot j_{10} + 1]$.)



Prior processing will have generated input arrays on 4 different instrument grids for each of the two views; the 1 km grid for the 3 thermal and fire channels, 0.5 km A and B grids for the solar channels, and a combined 0.5 km grid for the TDI channels.

The four instrument grids are as follows:

- 1) In terms of the notation defined in Section 4.2.3, the thermal and fire channel instrument grid is represented by the set of indices $S, p, k = 0, 1$, representing scan, pixel and detector number. Alternatively they may be indexed by scan trace and pixel index s, p , which is the approach adopted in Section 5.5.3.3.
- 2) The instrument grid for the A stripe (VIS channels and SWIR) is represented by the index set $S, p, t=0, 1, k = 0, \dots, 3$.
- 3) The instrument grid for the B stripe (SWIR only) by $S, p, t=0, 1, k = 4, \dots, 7$ (assuming the detector indexing illustrated in Section 5.5.3.2 (page 118). Again the ATBD adopts the combined indices $s = 4S + \text{row}(k), p' = 2p + t$. (The prime is then dropped for notational convenience).



4) *The TDI grid is that defined in Section 5.5.3.2.*

Each channel must be regridded into one of two output grids (at 1 km and 0.5 km resolution, as appropriate). There are two output grids for each view, which differ in that the grid for the solar channels has twice the dimensions of that for the solar thermal channels. The indices i and j refer to the appropriate array, as do the MAX and MIN limits. The 1.0 and 0.5 km image grids are coincident in the sense that alternate samples of the 0.5 km grid coincide with the samples of the 1.0 km grid. In detail, for each view we will have the following:

- *Three thermal and two fire channels, to be resampled from the 1 km instrument grid to the 1 km image grid.*
- *Six 'A stripe' solar channels to be regridded from the 'A stripe' instrument grid to the 0.5 km image grid (assuming that the visible channel detectors correspond to the A stripe).*
- *Three 'B stripe' SWIR channels to be regridded from the 'B stripe' instrument grid to the 0.5 km image grid (again assuming that the visible channel detectors correspond to the A stripe).*
- *Three combined TDI SWIR channels to be regridded from the TDI instrument grid to the 0.5 km image grid.*

ii) **Text on p 121 shown below was:**

'This algorithm must be modification, however, for the following reason. In order to retain the link between the regridded pixels and their original positions in the measurement geometry, the original instrument scan and pixel number are retained in an ADS, which is included in both the Level 1b and Level 2 full resolution products. If however these data are retained at full resolution, the resulting ADS are comparable in size to the Level 2 MDS, and contribute a disproportionate amount to the product size. For this reason a modified regridding algorithm was developed for AATSR that permitted the ADS to be sub-sampled to reduce its size.

For SLSTR the presence of the scan and pixel number ADS assumes increased importance because of its role in support of the Level 1c vegetation product. For reasons connected with the retention of orphan pixels, it is proposed to output the ADS at full resolution in the Level 1b product, but the problem of the size of the ADS in relation to the Level 2 product remains. It is therefore proposed to retain the AATSR modified algorithm to allow a sub-sampled ADS to be provided at Level 2. A further modification is required to ensure that orphan pixels are retained. The modified algorithm is set out in this section

*By way of clarification, we recall the following definitions. A **scan** refers to a single complete rotation of one or both of the nadir and oblique scan mirrors. The **scan index** S is an index to a synchronous pair of rotations of the two scan mirrors, as in Section 4.2.3.*

*The data associated with a scan comprises measurements for all detector elements. We define an **instrument scan** to refer to the ground trace associated with a single detector element. Instrument scans are identified by a single instrument scan number.*

In the case of the thermal channels, each channel will have two detector elements, indexed by $k = 0, 1$. The instrument scan number associated with a thermal channel instrument pixel will therefore be $2s + k + \sigma$, where σ is a constant integer offset chosen so that the instrument scan number is positive definite. In the case of the solar channels the detector index will take the values $k = 0, \dots, 3$ (we assume that TDI will have reduced the number of logical elements in the pixel groups from 8 to 4, corresponding to the four along-track positions of the detector array), and the corresponding instrument scan number will be $4s + k + \sigma'$ similarly.'



Text now:

'In order to retain the link between the regridded pixels and their original positions in the measurement geometry, the original instrument scan, pixel and detector number are retained in an ADS.'

iii) Text on p122

Text was:

'Initialize the regridded data arrays. This is necessary to ensure (a) that the requirement for cosmetic fill can be recognised; (b) that unfilled pixels can be identified at the conclusion of the regridding and cosmetic fill processes; and (c) that the regridded maximum and minimum temperature variables are initialized.

*nadir_fill_state(i, j) = UNFILLED_PIXEL
oblique_fill_state(i, j) = UNFILLED_PIXEL'*

Text now:

'Initialize the regridded data arrays. This is necessary to ensure (a) that the requirement for cosmetic fill can be recognised; (b) that unfilled pixels can be identified at the conclusion of the regridding and cosmetic fill processes; and (c) that the regridded maximum and minimum temperature variables are initialized.

*nadir_fill_state(i, j) = UNFILLED_PIXEL
oblique_fill_state(i, j) = UNFILLED_PIXEL*

These are the arrays which will contain the fill state of each pixel after regridding and cosmetic fill. The elements of these arrays may take one of 3 possible values:

*NATURAL_PIXEL The pixel has been assigned by the regridding algorithm
COSMETIC_PIXEL The pixel value has been assigned by the cosmetic fill algorithm
UNFILLED_PIXEL Neither the regridding nor cosmetic fill algorithm have assigned a value to this pixel*

Unfilled pixels occur because the cosmetic fill algorithm requires that an initially unfilled pixel can only be filled by an adjacent pixel. If a pixel has no naturally filled neighbours, it will remain unfilled. The pixel fill state will eventually appear in the product as an exception flag.

The fill values are used in the global confidence words in table of SY4 and the thermal and visible MDS exception words (see tables below).

Bit Number	Text Code	Description
0	<i>ISP_absent</i>	<i>ISP absent</i>
1	<i>pixel_absent</i>	<i>Pixel absent</i>
2	<i>not_decompressed</i>	<i>Not decompressed</i>
3	<i>no_signal</i>	<i>No signal in channel</i>
4	<i>saturation</i>	<i>Saturation in channel</i>
5	<i>invalid_radiance</i>	<i>Derived radiance outside calibration</i>
6	<i>no_parameters</i>	<i>Calibration parameters unavailable</i>



7	<i>unfilled_pixel</i>	<i>Unfilled pixel</i>
---	-----------------------	-----------------------

Bit	Text code	Meaning if set	Comment
0	<i>coastline</i>	<i>coastline in field of view</i>	
1	<i>ocean</i>	<i>ocean in field of view</i>	
2	<i>tidal</i>	<i>tidal zone in field of view</i>	
3	<i>land</i>	<i>land in field of view</i>	
4	<i>inland_water</i>	<i>inland water in field of view</i>	
5		<i>(spare)</i>	
6		<i>(spare)</i>	
7		<i>(spare)</i>	
8	<i>cosmetic</i>	<i>cosmetic fill pixel</i>	
9	<i>duplicate</i>	<i>duplicate pixel not regridded</i>	
10	<i>day</i>	<i>pixel in daylight</i>	
11	<i>twilight</i>	<i>pixel in twilight</i>	
12	<i>sun_glint</i>	<i>sun glint in pixel</i>	
13	<i>snow</i>	<i>snow</i>	
14	<i>summary_cloud</i>	<i>summary cloud test</i>	
15	<i>summary_pointing</i>	<i>summary pointing</i>	

iv) Text on p122

Text was:

```
'nadir_min_anc_temps(i, k) = +999.0 (k = 0, 1, ...5)
'nadir_max_anc_temps(i, k) = -999.0 (k = 0, 1, ...5)
'oblique_min_anc_temps(i, k) = +999.0 (k = 0, 1, ...5)
'oblique_max_anc_temps(i, k) = -999.0 (k = 0, 1, ...5)
'nadir_pixel_maps(i, 0) = -1
'nadir_pixel_maps(i, 1) = -1
'oblique_pixel_maps(i, 0) = -1
'oblique_pixel_maps(i, 1) = -1
'nadir_packet_invalid(i) = 0
'oblique_packet_invalid(i) = 0
```

Also initialise the instrument scan and pixel number to zero for all values of ig, j:

```
nadir_scan_number(ig, j) = 0
'nadir_pixel_number(ig, j) = 0
'oblique_scan_number(ig, j) = 0
'oblique_pixel_number(ig, j) = 0
```

Text now:

```
'nadir_min_anc_temps(i, k) = +999.0 (k = 0, 1, ...5)
'nadir_max_anc_temps(i, k) = -999.0 (k = 0, 1, ...5)

'oblique_min_anc_temps(i, k) = +999.0 (k = 0, 1, ...5)
'oblique_max_anc_temps(i, k) = -999.0 (k = 0, 1, ...5)
```



The above arrays hold the maximum and minimum detector temperatures recorded on the instrument scans that contribute to a given image row.

```
nadir_packet_invalid(i) = 0
oblique_packet_invalid(i) = 0
```

Invalidity flags for the source packets contributing to each pixel row.

Also initialise the instrument scan and pixel number to zero for all values of i, j :

```
nadir_scan_number(i, j) = 0
nadir_pixel_number(i, j) = 0
oblique_scan_number(i, j) = 0
oblique_pixel_number(i, j) = 0
```

These arrays are for use by level 1c and 2 processing to show the origin on the instrument grid of each pixel.'

i) Text on p 123 shown below to be deleted:

'If $i = 0$ (that is, if the value of i corresponds to a granule row), or this is the first scan to be regridded, or

```
nadir_scan_number(i_g, j) = 0
```

eq 5.5-3

go to **Step 5.5.2.4**. Otherwise test for modified regridding.

The following steps define a modified regridding scheme which permits the scan and pixel number ADS in the products to be subsampled in the along track direction. The present product definition proposes that the scan and pixel number ADS be presented at full resolution without subsampling. Consequentially the steps prior to **Step 5.5.2.4** are not required but are retained should the product definition be modified.'

ii) Text on p 124-5 shown below to be deleted:

'then go to **Step 5.5.2.4**.

Otherwise the natural regridding is not that required for the present granule. Try the other possibilities. Extract

$\delta i = \text{fractional part of } (\delta y)$ eq 5.5-4

$\delta j = \text{fractional part of } (\delta x)$ eq 5.5-5

Calculate

$\varepsilon_i = \text{sign}(\delta i - 0.5) \times 0.001 \varepsilon_y / \text{scale}$ eq 5.5-6

$\varepsilon_j = \text{sign}(\delta j - 0.5) \times 0.001 \varepsilon_x / \text{scale}$ eq 5.5-7

The constants ε_x and ε_y are the regridding tolerance thresholds taken from the configuration auxiliary file (RD-11 parameters `eps_x` and `eps_y`).

Step 5.5.2.1 Case <1> - x displacement

The regridding indices are re-calculated (i is unchanged)



$$\delta x = \text{scale}(\text{nadir_x_coord}(s, p'_n, k) + \varepsilon_j - \text{MIN_NADIR_X}) \quad \text{eq 5.5-8}$$

$$j = \text{FIX}(\delta x)$$

Test for modified regridting.

If

$$s_n = \text{nadir_scan_number}(i - i', j) + i'$$

and

$$p_n = \text{nadir_pixel_number}(i - i', j)$$

then go to **Step 5.5.2.4**.

Step 5.5.2.2 Case <2> - y displacement

The regridting indices are re-calculated. Find the (possibly new) index ig such that
 $\text{track_y}(ig + K) \leq \text{nadir_y_coord}(s, p'_n, k) + \varepsilon_i < \text{track_y}(ig + K + 1)$

Calculate

$$\Delta y = (\text{track_y}(ig + K + 1) - \text{track_y}(ig + K)) / \text{NGRANULE} \quad \text{eq 5.5-9}$$

$$\delta y = \text{scale}(\text{nadir_y_coord}(s, p'_n, k) + \varepsilon_i - \text{track_y}(ig + K)) / \Delta y \quad \text{eq 5.5-10}$$

$$i' = \text{integer part of } (\delta y) \quad \text{eq 5.5-11}$$

$$i = \text{scale} \cdot \text{NGRANULE} \cdot ig + i' \quad \text{eq 5.5-12}$$

$$\delta x = \text{scale}(\text{nadir_x_coord}(s, p'_n, k) - \text{MIN_NADIR_X}) \quad \text{eq 5.5-13}$$

$$j = \text{FIX}(\delta x) \quad \text{eq 5.5-14}$$

If $i' = 0$ (that is, if the value of i corresponds to a new granule row) go to **Step 5.5.2.4**. Otherwise test for modified regridting.

If

$$s_n = \text{nadir_scan_number}(i - i', j) + i'$$

and

$$p_n = \text{nadir_pixel_number}(i - i', j)$$

then go to **Step 5.5.2.4**.

eq 5.5-15

Step 5.5.2.3 Case <3> - displacement in both co-ordinates

The regridting indices are re-calculated (i is unchanged).

$$\delta x = \text{scale}(\text{nadir_x_coord}(s, p'_n, k) + \varepsilon_j - \text{MIN_NADIR_X}) \quad \text{eq 5.5-16}$$

$$j = \text{FIX}(\delta x) \quad \text{eq 5.5-17}$$

Test for modified regridting. If

$$s_n = \text{nadir_scan_number}(i - i', j) + i'$$

and

$$p_n = \text{nadir_pixel_number}(i - i', j)$$

then go to **Step 5.5.2.4**.

eq 5.5-18

If execution gets to here, this is an unused pixel. Set j to the next available orphan pixel index and continue.'

iii) Text from p 126 shown below to be deleted:

'go to **Step 5.5.3.4**. Otherwise test for modified regridting.

eq 5.5-19



The following steps define a modified regridding scheme which permits the scan and pixel number ADS in the products to be subsampled in the along track direction. The present product definition proposes that the scan and pixel number ADS be presented at full resolution without subsampling. Consequentially the steps prior to **Step 5.5.3.4** are not required but are retained should the product definition be modified.'

iv) Text from p 127-8 shown below to be deleted:

'Otherwise the natural regridding is not that required for the present granule. Try the other possibilities. Extract

$$\delta i = \text{fractional part of } (\delta y) \quad \text{eq 5.5-20}$$

$$\delta j = \text{fractional part of } (\delta x) \quad \text{eq 5.5-21}$$

Calculate

$$\varepsilon_i = \text{sign}(\delta i - 0.5) \times 0.001 \varepsilon_y / \text{scale} \quad \text{eq 5.5-22}$$

$$\varepsilon_j = \text{sign}(\delta j - 0.5) \times 0.001 \varepsilon_x / \text{scale} \quad \text{eq 5.5-23}$$

Step 5.5.3.1 Case <1> - x displacement

The regridding indices are re-calculated (*i* is unchanged)

$$\delta x = \text{scale}(\text{oblique_x_coords}(s, p'_a, k) + \varepsilon_j - \text{MIN_OBLIQUE_X}) \quad \text{eq 5.5-24}$$

$$j = \text{FIX}(\delta x) \quad \text{eq 5.5-25}$$

Test for modified regridding. If

$$s_a = \text{oblique_scan_number}(i - i', j) + i'$$

and

$$p_a = \text{oblique_pixel_number}(i - i', j)$$

then go to **Step 5.5.3.4**.

eq 5.5-26

Step 5.5.3.2 Case <2> - y displacement

The regridding indices are re-calculated. Find the (possibly new) index *ig* such that

$$\text{track_y}(ig + K) \leq \text{oblique_y_coord}(s, p'_a, k) + \varepsilon_i < \text{track_y}(ig + K + 1) \quad \text{eq 5.5-27}$$

Calculate

$$\Delta y = (\text{track_y}(ig + K + 1) - \text{track_y}(ig + K)) / \text{NGRANULE} \quad \text{eq 5.5-28}$$

$$\delta y = \text{scale}(\text{oblique_y_coords}(s, p'_a, k) + \varepsilon_i - \text{track_y}(ig + K)) / \Delta y \quad \text{eq 5.5-29}$$

$$i' = \text{intpt}(\delta y) \quad \text{eq 5.5-30}$$

$$i = \text{scale} \cdot \text{NGRANULE} \cdot ig + i' \quad \text{eq 5.5-31}$$

$$\delta x = \text{scale}(\text{oblique_x_coords}(s, p'_a, k) - \text{MIN_OBLIQUE_X}) \quad \text{eq 5.5-32}$$

$$j = \text{FIX}(\delta x). \quad \text{eq 5.5-33}$$



If $i' = 0$ (that is, if the value of i corresponds to a granule row) go to **Step 5.5.3.4**. Otherwise test for modified regridding. If

$$s_a = \text{oblique_scan_number}(i - i', j) + i'$$

and

$$p_a = \text{oblique_pixel_number}(i - i', j)$$

then go to **Step 5.5.3.4**.

eq 5.5-34

Step 5.5.3.3 Case <3> - displacement in both co-ordinates

The regridding indices are calculated (i is unchanged).

$$\delta x = \text{scale}(\text{oblique_x_coords}(s, p'_a, k) + \varepsilon_j - \text{MIN_OBLIQUE_X})$$

$$j = \text{FIX}(\delta x)$$

Test for modified regridding. If

$$s_a = \text{oblique_scan_number}(i - i', j) + i'$$

and

$$p_a = \text{oblique_pixel_number}(i - i', j)$$

then go to **Step 5.5.3.4**.

eq 5.5-35

If execution gets to here, this is an unused pixel. Set j to the next available orphan pixel index and continue.'

v) Text on p125

Add the following text to the beginning of step 5.5.2.4:

'Note that these steps should only be executed if the value of $\text{nadir_fill_state}[i, j]$ has the value UNFILLED_PIXEL . Otherwise the pixel and its co-ordinates should be saved as an orphan Pixel.'

vi) Text on p128

Add the following text to the beginning of step 5.5.3.4:

'Note that these steps should only be executed if the value of $\text{oblique_fill_state}[i, j]$ has the value UNFILLED_PIXEL . Otherwise the pixel and its co-ordinates should be saved as an orphan Pixel.'

vii) Text on p130

Text was:

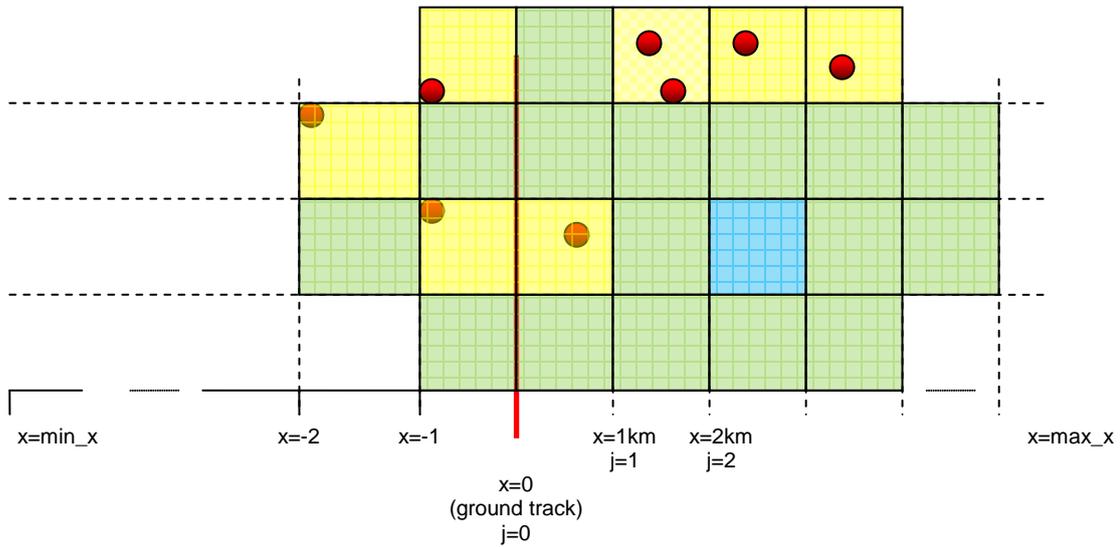
'This process is carried out on the regrided nadir image (view = nadir) and then on the regrided oblique image (view = oblique).'

Text (with diagram) now:

'This process is carried out on the regrided nadir image (view = nadir) and then on the regrided oblique image (view = oblique)., and is illustrated by the figure below:



Example of cosmetic fill for 1km image grid.



● Circles represent the x and y coordinates of the centre of the measurement pixels, x_n, y_n as previously defined

 Image grid square filled with measurement grid data or exception flag. **Naturally filled.**

 Image grid square filled with measurement grid data point from the nearest neighbour. **Cosmetically filled.**

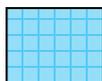
 Image grid square not filled. **Unfilled pixel.**

 Image grid square Naturally filled but there are two measurement pixels contained in the same square. One of these is an **orphan pixel.**

41. Section 5.5.2.8, p114

I have implemented meteo annotations in the SLSTR DPM. However, while working on it I discovered that meteo annotations in SY-4 are not fully consistent with the



ATBD scheme (OLCI ATBD copy paste). In the ATBD we first do a time interpolation at the mid-product time and then a bilinear interpolation in (lat, lon), but in SY-4 there are few variables that depend on time series. Except 'temperature_profile' and 'specific_humidity' variables that have a third additional vertical dimension, every variables should be 2D (i.e. the across track and along track dimensions), if we follow the ATBD algorithm, they should not depend on time.

Could you please clarify this point rapidly ?

Text was:

'Since the time sampling of meteo files do not match the Sentinel-3 acquisition, temporal interpolation is required and meteo files shall be available at the two closest time samples.'

Text now:

'Since the time sampling of meteo files do not match the Sentinel-3 acquisition, temporal interpolation is required and meteo files shall be available at the two closest time samples. Some of the fields require values before and after the nominal time in 6 hourly intervals for the diurnal thermocline modelling and so 5 values are required per analysis time i.e. -18, -12, -6, 0, +6hrs.'

42. Section 5.5.3.6.4, p142

2°) there is a conflict between ATBD and Product definition concerning the solar elevation angle.

According to the ATBD, the solar elevation angle is computed as $90 - \text{solar zenith angle}$. In step 5.5.3.6.4, 5.5.3.6.5 ... the solar elevation angle is indexed to the across-track band number. However in the Product Definition Document, the solar zenith depends on the across-track and along-track indexes.

Text was:

- a) For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are less than 5° : i.e. if
- b) $\langle \text{view} \rangle_{\text{band_centre_solar_elevation}}(i, 0) < 5.0$ or
- c) $\langle \text{view} \rangle_{\text{band_centre_solar_elevation}}(i, 9) < 5.0$,

Text now:

- a) For each scan i , retrieve the solar elevation at each end of the scan.

Calculate the solar elevation at each end of the scan using $90 - \text{solar zenith angle}$.

- The test is only performed if these are less than 5° : i.e. if
- b) $\langle \text{view} \rangle_{\text{solar_elevation}}(i, 0) < 5.0$ or
 - c) $\langle \text{view} \rangle_{\text{solar_elevation}}(i, \text{max_j}) < 5.0$,



43. Section 5.5.3.6.5, p143

2°) there is a conflict between ATBD and Product definition concerning the solar elevation angle.

According to the ATBD, the solar elevation angle is computed as $90 - \text{solar zenith angle}$. In step 5.5.3.6.4, 5.5.3.6.5 ... the solar elevation angle is indexed to the across-track band number. However in the Product Definition Document, the solar zenith depends on the across-track and along-track indexes.

Text was:

,

a) For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are less than 5° : i.e. if

b) $\langle \text{view} \rangle_{\text{band_centre_solar_elevation}}(i, 0) < 5.0$ or

c) $\langle \text{view} \rangle_{\text{band_centre_solar_elevation}}(i, 9) < 5.0$, '

Text now:

,

b) For each scan i , retrieve the solar elevation at each end of the scan.

Calculate the solar elevation at each end of the scan using $90 - \text{solar zenith angle}$.

The test is only performed if these are less than 5° : i.e. if

b) $\langle \text{view} \rangle_{\text{solar_elevation}}(i, 0) < 5.0$ or

c) $\langle \text{view} \rangle_{\text{solar_elevation}}(i, \text{max}_j) < 5.0$, '

44. Section 5.5.3.6.8, p150

2°) there is a conflict between ATBD and Product definition concerning the solar elevation angle.

According to the ATBD, the solar elevation angle is computed as $90 - \text{solar zenith angle}$. In step 5.5.3.6.4, 5.5.3.6.5 ... the solar elevation angle is indexed to the across-track band number. However in the Product Definition Document, the solar zenith depends on the across-track and along-track indexes.

Text was:

'The test is only applied to pixels on scans for which
 $\text{nadir_band_centre_solar_elevation}(i, 0) < 5.0$ and
 $\text{nadir_band_centre_solar_elevation}(i, 9) < 5.0$ and
 $\text{along_track_band_centre_solar_elevation}(i, 0) < 5.0$ and
 $\text{along_track_band_centre_solar_elevation}(i, 9) < 5.0$.'

Text now:

'The test is only applied to pixels on scans for which
 $\text{nadir_solar_elevation}(i, 0) < 5.0$ and
 $\text{nadir_solar_elevation}(i, \text{max}_j) < 5.0$ and
 $\text{along_track_solar_elevation}(i, 0) < 5.0$ and
 $\text{along_track_solar_elevation}(i, \text{max}_j) < 5.0$.'



45. Section 5.5.3.6.10, p161

2°) there is a conflict between ATBD and Product definition concerning the solar elevation angle.

According to the ATBD, the solar elevation angle is computed as $90 - \text{solar zenith angle}$. In step 5.5.3.6.4, 5.5.3.6.5 ... the solar elevation angle is indexed to the across-track band number. However in the Product Definition Document, the solar zenith depends on the across-track and along-track indexes.

Text was:

The following steps are applied for each view ($v = n | f$).

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

*`<view>_band_centre_solar_elevation(i, 0) ≥ 5.0 or
<view>_band_centre_solar_elevation(i, 9) ≥ 5.0,`*

Text now:

The following steps are applied for each view ($v = n | f$).

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

*`<view>_solar_elevation(i, 0) ≥ 5.0 or
<view>_solar_elevation(i, max j) ≥ 5.0,`*

46. Section 5.5.3.6.13, p163

2°) there is a conflict between ATBD and Product definition concerning the solar elevation angle.

According to the ATBD, the solar elevation angle is computed as $90 - \text{solar zenith angle}$. In step 5.5.3.6.4, 5.5.3.6.5 ... the solar elevation angle is indexed to the across-track band number. However in the Product Definition Document, the solar zenith depends on the across-track and along-track indexes.

Text was:

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

*`<view>_band_centre_solar_elevation(i, 0) ≥ 5.0 or
<view>_band_centre_solar_elevation(i, 9) ≥ 5.0,`*

Text now:

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

*`<view>_solar_elevation(i, 0) ≥ 5.0 or
<view>_solar_elevation(i, max j) ≥ 5.0,`*



47. Section 5.5.3.3, p125

For the time being, there is no cosmetic filling applied to the angles provided on the Tie Points grid since the sub-sampling is made before the cosmetic filling and the cosmetic filling only enables to fill missing solar angles on the full resolution grid as requested in the ATBD and not on the Tie Points grid. Do you confirm that this is exactly what is expected ?

Text was:

'Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled on the Tie Point grid for the product.'

Text now:

*'Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled on the Tie Point grid for the product **after the cosmetic fill has been performed.**'*

48. Section 5.5.3.3, p128

For the time being, there is no cosmetic filling applied to the angles provided on the Tie Points grid since the sub-sampling is made before the cosmetic filling and the cosmetic filling only enables to fill missing solar angles on the full resolution grid as requested in the ATBD and not on the Tie Points grid. Do you confirm that this is exactly what is expected ?

Text was:

'Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled on the Tie Point grid for the product.'

Text now:

*'Note that solar and viewing angles are computed for each instrument pixel and will be sub-sampled on the Tie Point grid for the product **after the cosmetic fill has been performed.**'*

Changes Added in Version 8 – 23 December 2010

49. Section 5.5.3.3, p123

Isabelle Muguet

Sent: Wed 24/11/2010 10:18

* Scan, pixel and detector arrays: I am not sure if the convention I adopted is the right one. In the ATBD, you seems to store in scan what you



defined as a scan trace (one for each detector row), and pixel is defined with an origin (i.e. "FIRST_NADIR_PIXEL_NUMBER") that is not provided in any ADF, and that I do not understand since the sampling depends on the channels used (PIX05SYNC for solar channels and PIX10SYNC for thermal ones).

first_nadir_pixel_number is provided in the ISPs are target_first_acquisition for the relevant view. The value will be different for the pix10 and pix05 channels. >>> To my understanding the acquisition number (i.e. pixel number) as defined in the IMDD is always incremented every PIX10SYNC whatever is the channels, and the acquisition in between at PIX05SYNC is handled in the ISPs using a cycle number 'C1' and 'C2'. Do you confirm? If confirmed, using scan, pixel and detector is not sufficient to uniquely map the regridded pixels with the instrument ones, since two successive acquisition of solar channels during one PIX10SYNC period will have exactly the same scan, pixel and detector indices. Thus, I assumed that pixel at the re-gridding stage refers to all acquisitions made even the ones that have been acquired every PIX05SYNC, as assumed in the TDI processing (p' combining p and t indices). In this case the first_acquisition_number stored in the ISPs is no more consistent with this convention. To be consistent with the Llc processing I suggest to add 2*FIRST_<view>_1km_PIXEL_NUMBER for all 0.5km grids in the DPM, instead of the first acquisition number, and to add FIRST_NADIR_1km_PIXEL_NUMBER, and FIRST_ALT_1km_PIXEL_NUMBER to the processing control parameters in the SY-4

Yes, the pixel number is incremented in 10pixsync for all channels. The pixel first acquisition that appears in the ISPs is indeed at pix10sync intervals and

FIRST_<view>_PIXEL_NUMBER = FIRST_<view>_1km_PIXEL_NUMBER

FIRST_<view>_05km_PIXEL_NUMBER = 2* FIRST_<view>_1km_PIXEL_NUMBER

Will add information to ATBD to clarify this.

Text was:

In Section 5.3.3.7 the x and y co-ordinates of each instrument pixel were determined. We introduce the notation $nadir_x_coord(s, p'_n, k)$, $nadir_y_coord(s, p'_n, k)$ for the x and y co-ordinates respectively of the nadir pixel identified by indices, s, p'_n, k . We also introduce MIN_NADIR_X , MAX_NADIR_X to denote the lower and upper limits of the instrument nadir swath in x units. Finally we define a factor $scale$, which takes the value 1 for the thermal channels and 2 for the solar channels (it is the reciprocal of the nominal resolution, in km.)

Text now:

In Section 5.3.3.7 the x and y co-ordinates of each instrument pixel were determined. We introduce the notation $nadir_x_coord(s, p'_n, k)$, $nadir_y_coord(s, p'_n, k)$ for the x and y co-ordinates respectively of the nadir pixel identified by indices, s, p'_n, k . We also introduce MIN_NADIR_X , MAX_NADIR_X to denote the lower and upper limits of the instrument nadir swath in x units. Finally we define a factor $scale$, which takes the value 1 for the thermal channels and 2 for the solar channels (it is the reciprocal of the nominal resolution, in km.) **The target first acquisition (FIRST_<VIEW>_PIXEL_NUMBER) from the instrument ISPs. This is incremented every PIX10SYNC, even for the solar and SWIR channels. The first pixel number of the solar and SWIR channels will therefore be twice that of the**



FIRST_<VIEW>_PIXEL_NUMBER provided in the ISP, and therefore the *scale* factor is used to account for this in the algorithms.

50. Section 5.5.3.3, p123

Isabelle Muguét

Sent: Wed 24/11/2010 10:18

* Scan, pixel and detector arrays: I am not sure if the convention I adopted is the right one. In the ATBD, you seems to store in scan what you defined as a scan trace (one for each detector row), and pixel is defined with an origin (i.e. "FIRST_NADIR_PIXEL_NUMBER") that is not provided in any ADF, and that I do not understand since the sampling depends on the channels used (PIX05SYNC for solar channels and PIX10SYNC for thermal ones).

first_nadir_pixel_number is provided in the ISPs are target_first_acquisition for the relevant view. The value will be different for the pix10 and pix05 channels. >>> To my understanding the acquisition number (i.e. pixel number) as defined in the IMDD is always incremented every PIX10SYNC whatever is the channels, and the acquisition in between at PIX05SYNC is handled in the ISPs using a cycle number 'C1' and 'C2'. Do you confirm ? If confirmed, using scan, pixel and detector is not sufficient to uniquely map the regrided pixels with the instrument ones, since two successive acquisition of solar channels during one PIX10SYNC period will have exactly the same scan, pixel and detector indices. Thus, I assumed that pixel at the re-gridding stage refers to all acquisitions made even the ones that have been acquired every PIX05SYNC, as assumed in the TDI processing (p' combining p and t indices). In this case the first_acquisition_number stored in the ISPs is no more consistent with this convention. To be consistent with the Llc processing I suggest to add 2*FIRST_<view>_1km_PIXEL_NUMBER for all 0.5km grids in the DPM, instead of the first acquisition number, and to add FIRST_NADIR_1km_PIXEL_NUMBER, and FIRST_ALT_1km_PIXEL_NUMBER to the processing control parameters in the SY-4

Equations 5-5-24 and 5-5-44

Text was:

FIRST_NADIR_PIXEL_NUMBER

Text now:

(*scale**FIRST_NADIR_PIXEL_NUMBER)

Equations 5-5-59 and 5-5-79

Text was:

FIRST_OBLIQUE_PIXEL_NUMBER

Text now:



(*scale**FIRST_OBLIQUE_PIXEL_NUMBER)

51. Section 5.5.3.7, p164

Isabelle Muguet

> Sent: 21 December 2010 13:21

> DEIMOS reported an important and urgent problem in the reflectance to
 > radiance conversion using the mean seasonally adjusted in-band solar
 > irradiance:
 >
 > the mean_solar_irradiance used during the VISCAL calibration
 > coefficients computation to compute the seasonally adjusted solar
 > irradiance (at step 7 of section 5.4.3.4. of the SLSTR ATBD) does not
 > depend on the detector index, whereas this is the case in the ADF
 > definition provided in SY-4 (table 6-39). Could you please clarify if
 > the mean seasonally adjusted in-band solar irradiance computed in this
 > step and used in section 5.5.3.7. of the ATBD should depend as well of
 > the detector index and in this case could you please update the
 > corresponding sections of the ATBD accordingly ?

Text was:

'to be the solar irradiance weighted by the SLSTR instrument passband.'

Text now:

'to be the solar irradiance weighted by the SLSTR instrument passband. It is noted that $E_{0,\lambda}$ is a function of view and detector number.'

Changes Added in Version 9 – 09-Jan-2011

52. Section 2.1 Applicable Documents

Phase A documents are not part of the current contract and should not be considered as ADs but may be considered as reference documents (RDs). As these are not referenced in the document, they have been removed from the document list.

Text was:

AD-1	Sentinel-3: Mission Requirements Document	EOP-SMO/1151/MD-md	6 December 2004
AD-2	GMES Sentinel-3 Definition Study System Requirements Document	EOP-FP/2004-07-953	17 December 2004

Text Now:

AD-1 Deleted reference – no longer applicable



AD-2 Deleted reference – no longer applicable

53. Section 2.1 Applicable Documents

Text was:

17 October 2007

Text now:

Issue 4.0, 13/11/09

54. Section 2.1 Applicable Documents

Text was:

AD-4	Earth Explorer Mission Software. EXPLORER_POINTING Software User Manual	EE-MA-DMS-GS-0006	19 January 2009
------	---	-------------------	-----------------

Text now:

AD-4	Earth Explorer Mission Software. EXPLORER_POINTING Software User Manual	EE-MA-DMS-GS-0005	Issue 4.1, 07/05/10
------	---	-------------------	---------------------

55. Section 2.2 Reference Documents

Text was:

RD-9	Data Handling and Mission Data Protocol Interface Specification	S3-IS-TAF-SC-00471
------	--	--------------------

Text now:

RD-9 Deleted reference – no longer applicable

56. Section 2.2 Reference Documents

Text was:

RD-11	SLSTR L1 Products Definition	S3-RS-RAL-SY-003	June 2010
-------	------------------------------	------------------	-----------

Text now:

RD-11	SLSTR L1 Products Definition	S3-RS-RAL-SY-003	Issue 5, 14/01/2011
-------	------------------------------	------------------	---------------------

57. Section 2.2 Reference Documents

Text was:

RD-4	Sentinel-3 Spacecraft Environmental and Test Requirements	S3-RS-AAF-SY-208	9 November 2007
------	--	------------------	-----------------



Text now:

RD-4 Sentinel-3 S3 Geometrical Performance SY-6 c, Issue 1, S3-TN- 20-Dec-2010
Budgets Part 1 : General rules and TAF-SY-01874
contributors

58. Section 2.2 Reference Documents

Consistent date formats used:

XX-Mon-XXX

59. Section 2.2 Reference Documents

Text was:

RD-10 GPP Perimeter and Mission Products S3-RP-GA-SL-00019
Format

Text now:

RD-10 GPP Perimeter and Mission Products S3-MO-TAF-00424/2008
Format

60. Section 2.2 Reference Documents

Text was:

RD-6 SLSTR Instrument Measurement Data S3-IS-GA-SL-00019
Definition Document

Text now:

RD-6 SLSTR Instrument Measurement Data S3-RP-GA-SL-00019
Definition Document

61. Header and footer throughout document

Text was:

Doc No: S3-SL-RAL-TN-032

Issue: 1.9

Date: 13-July-2010



©STFC 2010

Text now:

Doc No: S3-SL-RAL-TN-032

Issue: 2.0

Date: 14-Jan-2011

©STFC 2011

62. Section 4.4.3.1 , page 19

Text was:

This multiple is defined in the configuration ADF as `cycles_per_tie_point` (see section 6.8.3.1 of AD...).

Text now:

This multiple is defined in the configuration ADF as `cycles_per_tie_point` (see section 6.8.3.1 of [RD-11](#)).

63. Section 5.3.3.4 , page 70

Text was:

Because the leading edge of the nadir scan precedes the sub-satellite point, instrument scans measured before T_0 are necessary to fill in the image at the start of the orbit, and the origin of the look-up tables must precede T_0 by the interval $K\Delta T$, where K is an auxiliary constant whose value is TBD.

Text now:

Because the leading edge of the nadir scan precedes the sub-satellite point, instrument scans measured before T_0 are necessary to fill in the image at the start of the orbit, and the origin of the look-up tables must precede T_0 by the interval $K\Delta T$, where K is an auxiliary constant whose value is **to be found in the additional_scan_constant in the L1b processor ADF**.

64. Section 5.3.3.6 , page 70

Text was:

The following transformation matrices are independent of the scan position (section 4.4.4.74.4.4.7).

Text now:

The following transformation matrices are independent of the scan position (section [4.4.4.7](#)).

65. Throughout document

All figures numbered and referenced.

Auxiliary parameters used by the visible channel cloud test.



66. Section 4.4.4.10.4, page 41

Text was:

We now return to the solution of the triangle PQ_1Q_2 . Thus far our calculation has been exact, to the extent that sufficient terms of the expansion in (5.4.41)

Text now:

We now return to the solution of the triangle PQ_1Q_2 . Thus far our calculation has been exact, to the extent that sufficient terms of the expansion in (equation 5.4.46)

67. Section 5.5.3.5, page 132

Text was:

The surface classification flags are listed in Table 5-5 below and written to the global flags ADS.

Text now:

The surface classification flags are listed in Table 5-6 below and written to the global flags ADS.

68. Section 5.5.3.6, page 134

Text was:

The tests should be applied in the order in which they appear in Table 5-6

Text now:

The tests should be applied in the order in which they appear in Table 5-7

69. Section 5.5.3.6, page 98

Text was:

Scan Time Inconsistency Check

This checks that the time in the ISP is consistent with the scan count. Here we compare the actual time in the ISP against the expected time in the ISP based on the previous scan time.

```
Expected_time = time(scan-1) + 300ms.
```

```
ScanTimeErr = (ISP time stamp  $\neq$  Expected_time  $\pm$  80ms);
```

Text now:

Scan Time Inconsistency Check

This checks that the time in the ISP is consistent with the scan count. Here we compare the actual time in the ISP against the expected time in the ISP based on the previous scan time.

```
Expected_time = time(scan-1) + 300ms.
```

```
ScanTimeErr = (ISP time stamp  $\neq$  Expected_time  $\pm$  80ms);
```

Platform mode Check

The platform mode refers to the pointing guidance mode as contained in the NAVATT packets (see section 3.7 of RD-14) which is given once every second. Therefore, the pointing mode needs to be interpolated for each scan and then mapped onto the image grids during re-gridding.

70. Section 5.4.3.1.7.2, p166

Text was:

Auxiliary parameters used by the visible channel cloud test.



Parameter Name	Description	Comment
<u>N_VERT_v</u>	Number of vertices	
<u>N_ZONES_v</u>	Number of defined zones	
<u>X_v[j]</u>	X co-ordinate of vertex	$j = 0, \dots, N_{VERT} - 1$
<u>Y_v[j]</u>	Y co-ordinate of vertex j	$j = 0, \dots, N_{VERT} - 1$
<u>Vertex_index_v</u>	vertex index	$j = 0, \dots, N_{VERT} - 1$
<u>Zone_index_v</u>	Zone index	$i_zone = 0, \dots, N_ZONE - 1$.
<u>Vetex_id_v[k,i_zone]</u>	Vertex identifier,	$k = 0, 4$

(File will contain one record for each vertex $i = 0, \dots, N_VERT - 1$) There is provision for one set of vertices for each of 3 air mass ranges ($V=na, ob, int$ near-nadir, intermediate, along-track. The file will contain one record for each zone $i = 0, \dots, N_ZONE - 1$).

Modifications to the AATSR test: This is a single pixel test based on a surface classification. In AATSR processing the same classification LUT is used for both oblique and nadir views, so to this extent the test can be carried over unchanged. In principle slightly different tables could be used for the nadir and inclined view, and if this idea were adopted, we might also divide the nadir swath into zones as above by across-track limits X_1, X_2 such that:

- if $-X_1 < X < X_1$ the AATSR nadir view definitions are used;
- if $X_1 < \text{abs}(X) < X_2$, an intermediate set of definitions are used;
- of $X_2 < \text{abs}(X)$, the inclined view definitions are used.

The following steps are applied for each view ($v = n | f$).

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

- $\langle \text{view} \rangle_solar_elevation(i, 0) \geq 5.0$ or
- $\langle \text{view} \rangle_solar_elevation(i, \text{max } j) \geq 5.0$,

so excluding night-time data.

For each day-time pixel j identified above for which the visible channel reflectance values are valid, i.e. for which $I(ch, v; i, j) > 0$ for $ch = v870, v670, v555$ calculate the normalised difference indices as follows:

$$NDVI = (I(v870, v; i, j) - I(v670, v; i, j)) / (I(v870, v; i, j) + I(v670, v; i, j))$$

$$NDI2 = (I(v670, v; i, j) - I(v555, v; i, j)) / (I(v670, v; i, j) + I(v555, v; i, j))$$

Perform the following procedure to identify within which cluster defined by $NDVI$ and $NDV2$ the pixel falls.

For each zone $iz = 0, N_ZONE - 1$

Extract array of vertices:

$vertex[k] = vertex_id[k, i_zone], k = 0, 4.$

$N_SIDES = 5$

If $v[5] < 0$ then $N_SIDES = N_SIDES - 1$; (N_SIDES is the number of sides, equal to the number of vertices, of the polygon defining the zone. It will not be less than 4.)

Define arrays of dimension $N_SIDES + 1$ and extract the vertex co-ordinates into them:

For $k = 0, N_SIDES - 1$

$X[k] = X(k)$

$Y[k] = Y(k)$

end for

$X[N_SIDES] = X[0]$

$Y[N_SIDES] = Y[0]$

Identify whether the point PX, PY lies within this zone.

$FLAG = TRUE$

$PX = NDI2$

$PY = NDVI$

For $k = 0, N_SIDES - 1$

$$SX = X[k + 1] - X[k]$$

$$SY = Y[k + 1] - Y[k]$$



$$QX = PX - X[k]$$

$$QY = PY - Y[k]$$

Calculate the vector cross product $Q \times S$:

$$ZZ = (QX \times SY - QY \times SX)$$

$$FLAG = FLAG \text{ AND } (ZZ \geq 0)$$

If $FLAG = FALSE$ then exit loop; the point is not in this zone

End for

If $FLAG = TRUE$, exit loop ; we have found the zone

End for

The table of zones will only contain those regions that are associated with cloud. Thus the value of $FLAG$ at this point tells whether or not the pixel is cloudy.

If $FLAG = TRUE$ and the pixel is a land pixel for which the land flag (provided by the confidence parameter in the global flags) is set then flag the pixel as cloudy by setting the cloud flag vis_test for each view.

(Implementation note: although this algorithm in theory works over sea as well as land, use over sea is disabled for the time being. This is done in this step rather than by applying the test as a whole to land pixels only so that it can be easily enabled for sea pixels if desired at some time in the future.)

Text NOW:

Auxiliary parameters used by the visible channel cloud test are defined in the general parameters, zone vertices and zone definitions table in RD-11. These provide the information on the positions of the zones. There is provision for one set of vertices for each of 3 air mass ranges ($v=na$, int , ob (near-nadir, intermediate, oblique)).

Parameter Name	Description	Comment
N_SIDES	Number of vertices (or sides) per zone	
N_ZONES	Number of defined zones	
$Vertex_index[j_ver]$	Vertex index of vertex j_ver	$j_ver = 0, \dots, NSIDES - 1$
$Zone_index[i_zone]$	Zone index of zone i_zone	$i_zone = 0, \dots, N_ZONE - 1$
$Vetex_id[j_ver, i_zone]$	Vertex identifier	$j_ver = 0, \dots, NSIDES - 1, i_zone = 0, \dots, N_ZONE - 1$
$X_v[j_ver, i_zone]$	X co-ordinate of vertex j_ver in zone i_zone	$j_ver = 0, \dots, NSIDES - 1, i_zone = 0, \dots, N_ZONE - 1$
$Y_v[j_ver, i_zone]$	Y co-ordinate of vertex j_ver in zone i_zone	$j_ver = 0, \dots, NSIDES - 1, i_zone = 0, \dots, N_ZONE - 1$

Modifications to the AATSR test: This is a single pixel test based on a surface classification. In AATSR processing the same classification LUT is used for both oblique and nadir views, so to this extent the test can be carried over unchanged. In principle slightly different tables could be used for the nadir and inclined view, and if this idea were adopted, we might also divide the nadir swath into zones as above by across-track limits X_1, X_2 such that:

- if $-X_1 < X < X_1$ the AATSR nadir view definitions are used;
- if $X_1 < abs(X) < X_2$, an intermediate set of definitions are used;
- of $X_2 < abs(X)$, the inclined view definitions are used.

The following steps are applied for each view ($v = n | f$).

For each scan i , retrieve the solar elevation at each end of the scan. The test is only performed if these are not less than 5° : i.e. if

$$\langle view \rangle_solar_elevation(i, 0) \geq 5.0 \text{ or}$$



$\langle \text{view} \rangle_solar_elevation(i, \max j) \geq 5.0,$

so excluding night-time data.

For each day-time pixel j identified above for which the visible channel reflectance values are valid, i.e. for which $I(ch, v; i, j) > 0$ for $ch = v870, v670, v555$ calculate the normalised difference indices as follows:

$$NDVI = (I(v870, v; i, j) - I(v670, v; i, j)) / (I(v870, v; i, j) + I(v670, v; i, j))$$

$$NDI2 = (I(v670, v; i, j) - I(v555, v; i, j)) / (I(v670, v; i, j) + I(v555, v; i, j))$$

Perform the following procedure to identify within which cluster defined by $NDVI$ and $NDI2$ the pixel falls.

For each zone $i_zone = 0, N_ZONE - 1$

Extract array of vertices:

$Vertex[j_ver] = vertex_id[j_ver, i_zone], j_ver = 0, N_SIDES-1$

Define arrays (X_ar, Y_ar) of dimension $N_SIDES + 1$ and extract the vertex co-ordinates into them:

For each vertex $j_ver = 0, N_SIDES - 1$

$$X_ar[k] = X(k)$$

$$Y_ar[k] = Y(k)$$

end for

$$X_ar[N_SIDES] = X[0]$$

$$Y_ar[N_SIDES] = Y[0]$$

Identify whether the point PX, PY lies within this zone.

$FLAG = TRUE$

$PX = NDI2$

$PY = NDVI$

$$SX = X_ar[j_ver + 1] - X_ar[j_ver]$$

$$SY = Y_ar[j_ver + 1] - Y_ar[j_ver]$$

$$QX = PX - X_ar[j_ver]$$

$$QY = PY - Y_ar[j_ver]$$

Calculate the vector cross product $Q \times S$:

$$ZZ = (QX \times SY - QY \times SX)$$

$FLAG = FLAG \text{ AND } (ZZ \geq 0)$

If $FLAG = FALSE$ then exit loop; the point is not in this zone

End for