ThalesAlenia	Page de signatur	res électroniques / Electronic Signatures Page
		mentaire / Document Information
Auteur / Author : FR	NTINEL-3 LEVEL 1 ALGORITHMS T	
This document has been digi	ally signed and timestamped. To verify signa	tures validity, please refer to procedure and tools available on web site <u>pki.thalesaleniaspace.fr/pki/</u> gnature. After verification, the ? icon disappears if signature is valid. Last product update: july 2006.

Page laissée blanche intentionnellement

Blank page intentionally left



**REFERENCE:** S3-DD-TAF-SY-00620 **DATE:** 17/02/2015

**ISSUE:** 11 Page: 1/107

## Sentinel-3

## Level 1 Algorithms Theoretical Baseline Document - Part 2: Optical products [SY-24] Level 1c ATBD

Product Tree Code: 10 000 000

Written by	<b>Responsibility</b> + handwritten signature if no electronic workflow tool	Date
S3 TEAM		
Verified by		
J. RAVIOT LGM	Documentation officer	
D.LEBEDEFF	Optical Mission Performance	
J.L. PALMADE	Mission Manager	
Approved by		
Y. BAILLION	Project manager	
Reviewed by		
B. BERRUTI	ESA Project manager	

The validations evidence are kept through the documentation management system.



11

ISSUE:

Page: 2/107

## **DISTRIBUTION LIST**

TAS-F		
Yvan BAILLION	Χ	
Catherine EDERY		
Anne-Marie MONTET		
Florence CARRARO		Γ
Martine LAPOUGE		
Rodolphe KRAWCZYK		
Patrick NICOL		Γ
Frédéric JUCHET		Γ
Philippe SCHLOSSER		Γ
Valérie WEYNACHT		
Jean-Luc PALMADE	X	Γ
Alexandre HOUPERT		Γ
Dimitri LEBEDEFF	Х	Γ
François PAOLI	X	Γ
Michel VACANCE		Γ
Marc HERBELLEAU		Γ
Eric GARCIN		Γ
Robert KNOCKAERT		Ī
Gérard CLUZET		
Christophe GOUAULT		Γ
Jean-Christian NODET		Γ
Jean-Luc BEAUPELLET		Γ
Laurent SCARFO		
Stéphane BIANCHI	X	
François GRECO		
Yves DELCLAUD	X	
Franck GUERINEAU		Ī
Thierry GARNIER	x	
Dominique GIRAUD		
Thomas BLANCHARD		
Olivier CORRE		F
Gilbert VISSIO		F
Vincent COUPE		
Philippe GASNIER		F
Marc DESCHAUX BEAUME		F
Yves LE ROY		F
Patrice COSSARD		1
Jacques OSTER		
Olivier ZAGONI		

Arnaud BRETON	Γ	1
Olivier FAURE		1
Patrick INSALACO		1
Nicolas SERGENT		1
Jean-Bernard RITI		1
Mohamad AKOUM		1
		1
ESA		
Bruno BERRUTI	Х	
Marie-Hélène FERREIRA		
Constantin MAVROCORDATOS	Х	
Bernd SEITZ	Х	
Pierrik VUILLEUMIER		
		]
TAS-I Rome		1
Luciano ZUCCALA	1	1
TAS-I Milan		1
Luca MALTECCA		1
		1
TAS-E		1
José María RUIZ MINGUEZ		1
Rafael NAVARRO		1
Laurent BONNET		1
Alfonso RODRIGUEZ ALIJA		1
Juan CUETO RODRIGUEZ		1
Angel FERNANDEZ GONZALVO		
ALTER TECHNOLOGY		
Group Spain Olga RAMOS GARCIA		
Olga NAMOS GANCIA	-	
EADS CASA	+	
Raquel GONZALEZ	+	
	+	
DEIMOS	+	
Fabrizio PIRONDINI	+	
	+	
ACRI	+	
Odile FANTON D'ANDON	x	
	+ <u> </u>	

SELEX GALILEO	
Rubes VERATTI	Х
RAL	
Dave SMITH	X
JENA-OPTRONIK	
Wolfgang ENGEL	
CLS	
Ouan ZAN ZANIFE	
e2v	
Neil GUYATT	
SODERN	+
Didier LOISEAUX	+
SSBV	
Silvian KINDT	
VISTA	
Heike BACH	
ETCA	
Dominique DRUARD	
	1



## **ISSUE:** 11 Page: 3/107

## CHANGE RECORDS

ISSUE	DATE	§ CHANGE RECORDS	AUTHOR
Draft1	29/02/2008	First Draft	B.Francesconi
Draft2	30/05/2008	Overall revision	B.Francesconi
Draft3	02/07/2008	Mainly:	B.Francesconi
		* Title Change	
		* Deleted obsolete sections (resampling on an output grid)	
		* Replaced with new section 3.3	
		* Revision of section 1.4.2	
		* Revision of section 2	
		* Added details of interpolation algorithms (Annexes)	
		* Added details on dichotomic search (section 3.2.3.2.3)	
		* Added details on deforamtion models estimation (section 3.2.3.3)	
1	16/07/2008	Mainly:	B.Francesconi
		* "Algorithm Outputs" section added to each processing module	
		* "Processing parameters" sections added	
		* Intermediate outputs required for verification and analysis by the Processor	
		* Tie points rejection tests	
		* TOA radiance to TOA reflectance conversion algorithm for OLCI	
		* Misregistration for water pixels set to 0	
		* Other minor changes	
1.1	21/11/2008	Mainly:	B.Francesconi
		* Revision of section 2.	
		* Overall review of the deformation model (sections 3.2.4.3 and 3.2.2.4) and corresponding outputs (section 3.2.3)	
		* Making several notations and parameter names more explicit	
		* Other minor changes	
1.2	23/12/2008	* Introduction and description of the oblique view processing (mainly section 3.1)	B.Francesconi
		* Annotations to be included in the L1c product (sections 3.1 and 3.4)	
		* revision of Pre-processing (section 3.1):	
		- Description of the annotations retrieval processing	
		- No more coupled retrieval of ortho-rectified geolocation / misregistration data	
		- Conversion from TOA radiance to TOA reflectance	



**ISSUE:** 11 Page: 4/107

	1		1
		performed only for OLCI/SLSTR reference channels	
		- description of the algorithm to retrieve the OLCI and SLSTR nadir and along-track images in their acquisition geometry	
		* Better description of the module for computing the final correspondence grids (section 3.3)	
		* Revision and detailed description of the direct and inverse geolocation algorithms (section A.1)	
		* Other minor changes	
2	19/06/2009	* SLSTR Fire channels taken into account	B.Francesconi
		* updated OLCI channels (from updated SRD) taken into account	
		* "deregistration" -> "misregistration"	
		* "ortho-geolocation" -> "ortho-rectified geolocation" (from forthcoming SRD product definition)	
		* instrument description revised (from ACRI/RAL ATBDs)	
		* L1c product definition updated from latest information	
		* Several changes concerning pixels and annotations retrieval from L1b products (§3.1.4.2 and 3.1.4.3). But this processing still needs improvements and clarifications.	
3	13/11/09	Overall update according to the more up to date SY-4 volumes and OLCI/SLSTR ATBDs	B.Francesconi
		* Reorganization of section 1 (Introduction). Instruments description has been reduced to minimum, now referring to OLCI and SLSTR ATBDs.	
		* Minor changes in section 2	
		* Section 3.1 (Pre-processing):	
		<ul> <li>Details on the algorithms to retrieve the images in acquisition geometry and their annotations have been added. SLSTR 1km and 500m channels are now distinguished. SLSTR channels are cut.</li> </ul>	
		- Conversion to TOA reflectance detailed	
		- Tie-Point selection fully described	
		* Section 3.2 (Misregistration Estimation): Minor changes	
		- Changes in Algorithm inputs	
		<ul> <li>Switches allowing to perform or not several steps have been added</li> </ul>	
		* Section 3.3 (Correspondence computation):	
		- Clarifications of inputs and algorithms	
		<ul> <li>Cutting of the SLSTR nadir and oblique view images to fit OLCI swath and span</li> </ul>	
		<ul> <li>Detailed description of the SLSTR nadir view collocation grids. 1km and 500m channels are now distinguished</li> </ul>	
		* Section 3.4 (Annotations): Cleaning of the section, referring to L1c product definition (SY-4, volume 4)	



**ISSUE:** 11 Page: 5/107

		* Small changes in ANNEX 1	
4	22/01/2010	Main changes concern:	B.Francesconi
4	22/01/2010	* Section 3.1 (Pre-processing):	D.I Tancescom
		<ul> <li>Major changes in SLSTR product pre-processing due to a major update of SLSTR product definition (SY-4, volume 3, issue 4.3</li> </ul>	
		<ul> <li>Management of SLSTR SWIR sub-band selection ("A", "B" o TDI)</li> </ul>	
		<ul> <li>Tie points selection (§ 3.1.4.6): a processing for selecting regularly spaced tie-points in the OLCI image has been added as a user-switchable alternative to the use of the tie- points database</li> </ul>	
		* Section 3.2 (Misregistration Estimation):	
		- Tie points coordinates can now be taken in an input file	
		<ul> <li>Additional tests during the construction of the Context and Search imagettes are described</li> </ul>	
		- The algorithms to compute the $G_{12}$ and $\hat{G}_{12}$ correspondence grids are now described (§3.2.4.1.2 and § 3.2.4.3.3)	
		- Tie-points rejection tests added (§3.2.4.2.2)	
		- Range of the estimated deregistration shifts is now limited	
		* Section 3.3 (Correspondence computation):	
		<ul> <li>Management of SLSTR SWIR sub-band selection ("A", "B" o TDI)</li> </ul>	
		<ul> <li><u>The algorithm to compute the O<sub>g</sub> / S<sub>b</sub> correspondence grids</u> is now described (§ 3.3.4.3) and in accordance with Auxiliary data described in SY-4 volume 4</li> </ul>	
		* Section 3.4 (Annotations):	
		- Small update according to SY-4 volume 3 (SLSTR) update	
5	17/06/2010	Overall update according to: * latest <i>SLSTR Product definition</i> (SY-4 volume 3, issue 4.4 draft G, 28/05/2010) and ATBD (Issue 1.8 draft F, 14/06/2010). No major changes are expected in their final issues. * latest <i>OLCI Product definition</i> (SY-4 volume 2, issue 5 rev0) and ATBD (Issue 2.1, 21/05/2010). * ACRI-ST and DEIMOS feedback, during O-GPP development. This implies clarifications/corrections throughout the document.	B.Francesconi
		Main changes concern:	
		* Section 3.1 (Pre-processing):	
		<ul> <li>Clarification of several definitions &amp; and update of references to notations/definitions in OLCI and SLSTR SY- 24 and SY-4</li> </ul>	
		<ul> <li>Major changes in SLSTR product pre-processing due to a major update of SLSTR product definition (use of a ortho- rectified geolocation file now included in SLSTR product).</li> </ul>	



### **REFERENCE:** S3-DD-TAF-SY-00620 **DATE:** 17/02/2015

**ISSUE:** 11 Page: 6/107

1			
		Deletion of § 3.1.4.3 ("Retrieve the ortho-rectified geolocation grids of the nadir and along-track view SLSTR channels") and adaptation of new § 3.1.4.4	
		<ul> <li>Handling of additional SLSTR annotations (quality information) following RAL recommendation</li> </ul>	
		<ul> <li>Change in § 3.1.4.4 (TOA radiance conversion) according to memo S3-MO-TAF-01141/2010 ("Addendum to L1c ATBD (S3-DD-TAF-SY-00620 Issue 4): Modification of formulae for TOA reflectance conversion").</li> </ul>	
		* Section 3.2 (Misregistration Estimation):	
		- Correction of eq. 3-4, eq. 3-15, eq. 3-16	
		<ul> <li>In § 3.2.4.2.3.2, overall clarifications added and modification of the second stop criteria after DEIMOS found it sometimes irrelevant</li> </ul>	
		* Appendix A.1 (Direct ortho-rectified geolocation function)	
		<ul> <li>The altitude is now an output of this function (required for processing)</li> </ul>	
6	30/07/2010	This issue has been finalized from 2 draft issues delivered to ACRI and described here below for memory:	B.Francesconi
		Draft1(22/07/2010):	
		Correction of the algorithms in section 3.3.4.5.1 and 3.3.4.6 due to an unexpected change in the SLSTR design (the scan rotation direction has changed). The processing is now able to any scan rotation direction (East to West or West to East).	
		Typos correction	
		Draft2 (30/07/2010):	
		Changes according to ACRI-ST and DEIMOS feedback. Mainly:	
		<ul> <li>Correction in section 3.3.4.5 (OLCI and oblique view collocation grid on common swath) to avoid errors when calling the inverse geolocation function for OLCI pixels on the edge of common swath</li> </ul>	
		Minor change in Inverse geolocation function (Annex A)	
		Typos/format correction	
7	14/01/2011	Minor corrections and clarifications according to ACRI-ST and DEIMOS feedback.	B.Francesconi
		Typos correction	
		Update of reference document issues	
8	22/07/2011	Change in section 3.1.2.1 (Retrieving the 5 camera module images in their acquisition geometry) according to ACRI clarifications	B.Francesconi
		Reduction of the number of TBD/TBC and assumptions (answer to CDR RID OP-13)	
		Other minor changes for clarifications and typos corrections	



## REFERENCE: S3-DD-TAF-SY-00620 DATE: 17/02/2015

**Issue:** 11 Page: 7/107

9	27/04/2012	Update for O-GPP V2 according to latest L1c, OLCI and SLSTR Product definitions + OLCI and SLSTR ATBDs	B.Francesconi
		Complete revision of the oblique view collocation algorithm to cope with computation time issues (convergence issue of the previous algorithm due to interleaved scan traces in the oblique view images).	
		This processing is now performed before "Selecting the part of the SLSTR nadir view image covered by OLCI image" (§3.3.4.5)	
		This also implies small changes in the pre-processing stage.	
		Other minor changes in pre-processing following change in SLSTR L1b product.	
		Corrected 2 rejection tests (T_MAXMEAN_DIFF_COR and T_MAXMAX_DIFF_COR) on the correlation surface to avoid adverse behavior in certain conditions.	
		FIRST_NADIR_1km_PIXEL_NUMBER and FIRST_ALONG_TRACK_1km_PIXEL_NUMBER parameters now read directly in SLST L1b product (formerly input from processing control ADF)	
		Added some clarifications/typo corrections where needed	
10	11/06/2013	§ 2.2: clarification that the OLCI/SLSTR reference bands are given as input parameters.	B.Francesconi
		§ 3.1.1.3: Processing parameters added	
		Added § 3.1.4.1 specifying that tests on the consistency of input parameters shall be done at the very beginning of the L1c processing. Several tests added or moved to this section.	
		§ 3.1.4.3.2: Added test for presence of TDI channels in L1b products in	
		§ 3.3.4.6: typo correction: The test "IF Cm36(k,j)≠(ORPHAN, ORPHAN)" replaced by ≠(NO_CORRESP_VALUE, NO_CORRESP_VALUE)	
		Correction of typos in Figure 3-9	
		§ 3.3.1.6: Processing parameter added.	
11	17/02/2015	§ 1.2.1: Reference Documents update.	D. Lebedeff
		§ 2.1.1: SLSTR L1b TDI bands are optional since V2.	
		§ 3.2.1.3: Parameter CW_J_RADIUS has been removed as not used since V2.	
		§ 3.2.4.1.1: Normalisation of rejection test CW_QT_5 has been removed.	
		§ 3.2.4.1.2: Normalisation of rejection test SW_QT_4 has been removed.	
		§ 3.2: New tie points rejection test that applies on subpixel shifts.	



**ISSUE:** 11 Page: 8/107

## TABLE OF CONTENTS

1.	INTI	RODUCTION1	1
1.	1	Document Context and Scope	11
1.	2	Relevant Documents	11
	1.2.1		
	1.2.2		
	1.2.2		
1.	3	Definitions, Acronyms and Notations	12
	1.3.1	Notations	12
	1.3.2		
	1.3.3	Acronyms	15
1.	4	Overview of instruments and data acquisition baseline	16
	1.4.1		
	1.4.2	OLCI and SLSTR acquisition scheme and pixel numbering	18
2.		EL 1C OVERVIEW AND BACKGROUND INFORMATION	
2.	1 2.1.1	Synergy Product Definition	
	2.1.1		
	2.1.2		21
2.	2	Objectives and outlines of Level 1c Processing	21
	-		
3.	ALG	ORITHMS DESCRIPTION	23
3.	1	Pre-processing	22
5.	3.1.1		
		1.1.1 OLCI FR L1b product and SLSTR L1b product	
		1.1.2 A tie points database	
		1.1.3 Processing parameters	
	3.1.2	Processing Objective	25
		1.2.1 Retrieve the full images of the 5 OLCI camera modules in their acquisition geometry from L1b	
		ta, and their annotations	25
	-	1.2.2 Retrieve the channels of the SLSTR nadir and oblique view in their acquisition geometry from	
		b data, and their annotations	25
		1.2.3 Convert the OLCI and SLSTR nadir view reference channels from TOA radiance to TOA	25
		flectance	
	3.1.3		
	3.1.4	<b>5 1</b>	
		1.4.1 Check inputs	
		1.4.2 Retrieve the full images of the 5 OLCI camera modules in their acquisition geometry from L1b	
		Ita, and their annotations	
	3.1	1.4.3 Retrieve the SLSTR channels in their acquisition geometry from L1b data, and related per pixe	
		inotations	
		1.4.4 Retrieve the Sun Zenith Angle (SZA) for all instrument pixels the nadir view SLSTR S <sub>u</sub> channel	
		1.4.5 Convert the radiometric unit of OLCI and SLSTR nadir view reference channels	
		1.4.6 Tie points selection	
	3.	1.4.6.1 Selection of tie-points from the tie-points database:	44



## REFERENCE: S3-DD-TAF-SY-00620 DATE: 17/02/2015

**ISSUE:** 11 Page: 9/107

3.1.4.6		
3.1.4.6	gins (m = 1 to 5) 3 Selection of tie-points on a regular grid within the image part of each O <sub>a</sub> <sup>m</sup> image defined	46 by
	s (m = 1 to 5)	
3.1.4.6.		49
2.2 Inter	Instrument Spatial Microsistration Estimation, OLCL/SLSTB Nadix view	40
	Instrument Spatial Misregistration Estimation: OLCI / SLSTR Nadir view	
3.2.1 A	One SLSTR nadir band $S_u$ and one OLCI FR band $O_q$ selected as reference for compariso	n 50
3.2.1.1	A list of N_TP_L1C(m) tie points coordinates per OLCI camera module m	50
3.2.1.2	Processing parameters	
	ocessing Objective	
3.2.2.1	Extraction of a couple of geometrically normalized OLCI $O_q$ / SLSTR $S_u$ imagettes around t	he tie
point p	51	
3.2.2.2	Radiometric Normalization (not needed in baseline)	
3.2.2.3	Sub-Pixel Local Shift Estimation at Tie Point by Matching Context and Search imagettes	
3.2.2.4	Estimation of Deformation Model Parameters	
	gorithm Outputs	
	athematical Description	
3.2.4.1	Extraction of a geometrically normalized OLCI O <sub>g</sub> / SLSTR S <sub>u</sub> imagettes couple around the	tie
point p	53	
3.2.4.2	Local Shifts Estimation by Matching Context and Search Windows	61
3.2.4.3	Deformation Model Estimation	67
3.3 Com	puting the correspondence between OLCI Oq pixels and all other SLSTR and OLCI char	<mark>าnels78</mark>
	gorithm Inputs	
3.3.1.1	OLCI inter-channel spatial misregistration Auxiliary Data File	
3.3.1.2	SLSTR nadir view inter-channel spatial misregistration Auxiliary Data File	78
3.3.1.3	SLSTR S <sub>u</sub> (nadir view) / OLCI Oq <sup>m</sup> inter-instruments misregistration grids	79
3.3.1.4	OLCI Oq <sup>m</sup> ortho-rectified geolocation grids	79
3.3.1.5	SLSTR oblique view ortho-rectified geolocation grids for 500m and 1km channels	
3.3.1.6	Processing Parameters	
	ocessing Objective	
	gorithm Outputs	
	athematical Description	
3.3.4.1	Establishing the correspondence between OLCI $O_{q}^{m}$ and OLCI $O_{b}^{m}$ channels (b $\neq$ q)	82
3.3.4.2	Establishing the correspondence between OLCI $O_q^m$ and SLSTR $S_u$ channel	82
3.3.4.3	Establishing the correspondence between OLCI $O_q^{im}$ and SLSTR S <sub>b</sub> , channels (b' $\neq$ u)	83
3.3.4.4	Establishing the collocation grids between OLCI O <sup>im</sup> and SLSTR oblique view channels	
3.3.4.5	Selecting the part of the SLSTR nadir view image covered by OLCI image	
3.3.4.6	Selecting the part of the SLSTR oblique view image covered by OLCI image	91
3.4 Con	struction of L1c annotations	03
	gorithm Inputs	
3.4.1.1	OLCI and SLSTR L1b Annotations	
3.4.1.2	L1c OLCI/SLSTR mis-registration and collocation grids	
3.4.1.3	L1c Tie- Points statistics.	
	ocessing Objective	
	gorithm Outputs	
	athematical Description	
3.4.4.1	L1c annotations from L1b OLCI annotations.	
3.4.4.2	L1c annotations from SLSTR annotations	
3.4.4.3	L1c specific annotations	



**Issue:** 11 Page: 10/107

#### **APPENDIX A - COMPUTATIONAL DEFINITION OF DIRECT AND INVERSE ORTHO-**A.1 A.1.1 An ortho-rectified geolocation grid ......97 A.1.1.1 A.1.1.2 A.1.1.3 A.1.2 A.1.3 A.2 A.2.1.1 A.2.1.2 A.2.1.3 A.2.1.4 A.2.2 A.2.3 Β. **B.1 B.2 B.3 B.4 C.1 C.2 C.3 C.4**



## 1. INTRODUCTION

## 1.1 Document Context and Scope

This Algorithm Theoretical Basis document (ATBD) describes the algorithms used to produce the Level 1c Synergy Product from OLC and SLST L1b products. For each processing module the list of inputs and outputs is given. A list of intermediate outputs is also given for verification purpose (to be used by the O-SPS for instance).

## **1.2 Relevant Documents**

## 1.2.1 Reference Documents

Reference Documents (RD) do not contain requirements applicable to the user of this document. They are listed for traceability reasons only, as they have been used during the preparation of this document.

The following documents have been used to prepare this document.

RDXXX means ESA reference

RD-XXX means TAS-F reference

	TITLE	REFERENCE
AD03	GMES Sentinel-3 System Requirements	S3-RS-ESA-SY-0010
	Document	Issue 4, 13/11/2009
AD07	GMES Sentinel-3 Phase-B2/C/D/E1 Document	S3-LI-ESA-SY-0005
	Requirement Description (DRD)	Issue 3.0, 15/10/2007

	TITLE	REFERENCE
RD-1	OLCI Level 0, Level 1b Algorithm Theoretical	S3-ACR-TN-007
	Basis Document	Issue 5.0, 10/12/2014,
RD-2	SLSTR: Level 1 Algorithm Theoretical Basis	S3-SL-RAL-TN-32
	Definition Document for Level 1 Observables	New Issue for O-GPP V3
RD-3	Level 0, Level 1a/b/c Products Definition	S3-RS-RAL-SY-00003
	Part 2: Optical Products	New Issue for O-GPP V3
	Volume 3: SLSTR products	
RD-4	Level 0, Level 1a/b/c Products Definition	S3-RS-ACR-SY-00004
	Part 2: Optical Products	New Issue for O-GPP V3
	Volume 2: OLCI products	
RD-5	Level 0, Level 1 Products Definition	S3-RS-TAF-SY-01247
	Part 2: Optical Products	<mark>lssue 10, To be </mark> issued
	Volume 4: Level 1C product	
RD-6	Level 1c deformation model: Justifications and	S3-MO-TAF-00454/2008
	Trade-off analysis	02/12/2008
RD-7	Loss of pixels on regridding in SLSTR Level 1B	S3-SL-RAL-TN-048
	Processing	June 2008



**ISSUE:** 11 Page: 12/107

### 1.2.2 Relevant Publications

- [DR04] Delon, J, B. Rougé, 2004. *Analytical study of the stereoscopic correlation*. Technical report n°19. Centre des Mathématiques Appliquées. Ecole Normale Supérieure de Cachan. 21 pp.
- [ZF03] Barbara Zitová, Jan Flusser, 2003, *Image registration methods: a survey*, Image and Vision Computing 21 (2003) pp. 977–1000
- [S07] Schowengerdt, R. A., 1997, *Remote sensing: models and methods for image processing*, Academic Press (Editor)
- [SA03] S. Sylvander, I. Albert-Grousset, P. Henry, 2003, *Geometrical Performances of the VEGETATION Products*, IGARSS 2003
- [EL07] Eastman, Roger D.; Le Moigne, Jacqueline; Netanyahu, Nathan S., *Research issues in image registration for remote sensing*, CVPR '07. IEEE Page(s):1 8
- [GV96] Gene H. Golub, Charles F. Van Loan, *Matrix computations (3rd ed.)*, November 1996, Johns Hopkins University Press
- [B99] Blanc, Philippe, *Développement de méthodes pour la détection de changement*, 1999, PhD Thesis
- [W90] Wahba, Grace, *Spline Models For Observational Data*, 1990, Society for industrial and applied mathematics, Philadelphia
- [K81] R. Keys, *Cubic convolution interpolation for digital image processing*, 1981, IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing 29: 1153

### **1.3 Definitions, Acronyms and Notations**

### 1.3.1 Notations

(k,j) or (k',j') or (k',j'): image coordinates (row, column) (can be non-integer)

(x,y,z) or (x',y',z'): terrain coordinates (in any coordinates system)

 $(\lambda, \phi, h)$ : latitude, longitude, altitude. Terrain coordinates in geographic system.

 $\hat{X}$  is an estimation of the value of *X* , resulting from processing

FLOOR[X]: the largest integer not greater than X (or the integer part of X)

round(x): the integer that is closest to x

abs(x): the absolute value of x

max(x): the maximum value of x (x can be a set or a function for instance)

 $\circ$ : f<sub>2</sub>  $\circ$  f<sub>1</sub> is the mathematical composition of functions f1 and f1: f<sub>2</sub>  $\circ$  f<sub>1</sub>(x) = f<sub>2</sub>(f<sub>1</sub>(x))

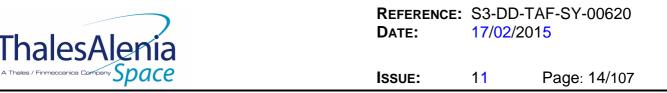
x mod y is the remainder of division of x by y



 $\|\mathbf{X}\|$  is the Euclidian norm of vector  $\mathbf{X}$ 

## 1.3.2 Definitions

- Solar channels: Channels with centre wavelength lower than 3.0 μm (SLSTR S1 to S6 and all OLCI channels)
- Thermal channels: Channels with centre wavelength larger than 3.0  $\mu m$  (SLSTR S7 to S9 and F1, F2)
- **Visible** radiation: electromagnetic radiation detectable by the human eye with a wavelength between approximately 400nm and 700nm (OLCI Oa1 to Oa11 and SLSTR S1 and S2 channels)
- Infra-red (IR) radiation: electromagnetic radiation of wavelengths between about 750nm and 1mm. This is broken down into 5 wavelength regions
   Near-IR 0.75-1.4 μm (OLCI Oa12 to Oa21, SLSTR S3 and S4 channels)
   Short-Wave IR 1.4-3 μm (SLSTR S5 and S6 channels)
   Medium-Wave IR 3-8 μm (SLSTR S7 and F1 channels)
   Long-Wave IR 8–15 μm (SLSTR S8, S9 and F2 channels)
- **Tie point** or **Correlation point**: landmark, visible and located on two images, where local residual misregistration between these images is estimated by a matching process.
- **Earth Surface**: The Earth surface is modeled as a Digital Elevation Model (provided as CFI) on top of the WGS84 ellipsoid model.
- **(Direct) Geolocation function**: function that maps a point (k,j) (possibly non-integers) in an image to a point (x,y,z) *on the ellipsoid surface*. It is subtended by a model of the line of sight coming from point (k,j).
- Inverse Geolocation function: the inverse function of the direct geolocation function.
- (Direct) Ortho-rectified geolocation function: function that maps a point (k,j) (possibly non-integers) in an image to a point (x,y) on the Earth surface, by taking into account a Digital Elevation Model (DEM) z = DEM(x,y). Theoretically (x,y,z) is the intersection of the line of sight coming from (k,j) with the Earth surface modelled as a DEM on top of a reference ellipsoid. The terrain point location is corrected from the relief effect, compared to the one computed with the direct geolocation function.
- **Inverse Ortho-rectified geolocation function**: the inverse function of the direct orthorectified geolocation function.
- **Restituted value**: value retrieved when all known corrections have been applied.
- (Mis-)Knowledge Error: residual error when all known corrections have been applied. The true value is given by adding the (unknown) (Mis-)Knowledge Error to the restituted value.
- Inter-channel spatial co-registration, simply referred here as co-registration or misregistration: The definition given in [AD03] is: *Maximum equivalent ground distance between the positions of all pairs of spatial samples acquired in two spectral channels and related to the same target on Earth.*
- Inter-instrument spatial misregistration: misregistration between one reference OLCI channel and one reference SLSTR channel.
- Intra-instrument spatial misregistration: misregistration between all the channels within a same instrument.



- **Image Sample/Pixel**: Pixel stands for **Pi**cture **El**ement. Each *pixel* is a measure of *radiance* generally gridded, with coordinates (k,j) in an image. k indexes the rows, j indexes the columns.
- **Instrument sample / Instrument pixel / Acquired pixel**: (all equivalent terms). Pixels really acquired by an instrument, before any geometric transformation.
- Frame: the set of measurements acquired by the OLCI instrument at a given time
- Coastal zone: Sea surface extending from the coast up to 300km offshore (from [AD-1])
- Ancillary data: A classical definition is "All on-board data, other than Observation and HKTM data, necessary for the Products processing". This would include in particular not only various parameters and settings but also satellite data such as OBT and Time correlations if needed, Navigation data, etc.
- Auxiliary data: We limit our understanding of Auxiliary data to all complementary data provided to the Ground Segment (PDGS) by external providers in order to process the Level 1 and above products.
- **Product Data**: Any data produced by the Ground-Segment Processing
- **Search window**: small window (grid) centered on a tie-point in OLCI geometry. Thus it is a set of coordinates. It is used to extract a search imagette of SLSTR channel for correlation with the Context window during the inter-instrument misregistration estimation.
- **Context window**: small window (grid) moved around the Search Window (along shift vectors) in OLCI channel geometry. Thus it is a set of coordinates. It is used to extract a context imagette of OLCI channel for correlation with the search chip during the inter-instrument misregistration estimation
- **Context imagette**: It is the radiometric counterpart of the context window, obtained by extracting the OLCI channel radiometry corresponding to the context window. If C is a Context imagette W(C) represents the corresponding Context window
- Search imagette: It is the radiometric counterpart of the search window, obtained by resampling the SLSTR channel radiometry to the search window. If S is a Search imagette W(S) represents the corresponding Search window
- Acquisition geometry: See sections 3.1.4.2.1 and 3.1.4.3.1.
- **Orbital Revolution Number**: This number identifies the Sentinel 3 orbit within the orbital cycle. There are 385 orbits per cycle, thus the Orbital Revolution Number is between 0 and 384.
- Orphan pixels or Removed pixels: These are pixels acquired by the instruments but not retained in the L1b gridded image, due to the L1b (nearest neighbor) projection on the product grid. For OLCI, those pixels mainly come from overlapping areas between adjacent camera modules. For SLSTR they may come from a possible oversampled acquisition at nadir of the nadir-view, with respect to the L1b image gridding (see [RD-7]). In oblique view there are many orphans due to scan-to-scan along-track overlap. To answer the L1c requirements, all those pixels are retained in L1b products but not gridded. Note: The expression "orphan pixel" is used in SLSTR documents ([RD-2], [RD-3]) while "removed pixel" is used in OLCI documents ([RD-1], [RD-4]). The term "ungridded pixels" is also used instead of orphan or removed pixels in this document.
- Scan: A scan is defined as a complete rotation of the SLSTR scan mirrors.
- **Instrument scan or scan trace**: It is the trace of a single SLSTR 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.



- **ISSUE:** 11 Page: 15/107
- **Image scan**: It is a line of pixels in the SLST L1b product. Note that in the L1c product an *image scan* and an *instrument scan* should refer to the same thing.
- **Deformation model**: In the L1c processing this terms refers to the interpolation model applied on the (potentially) irregular grid of tie-points and representing the deformation field between OLCI and SLSTR in the OLCI geometry
- **Correspondence grids**: These grids are the main output of the L1c processing, stored in the Misregistration datafiles in the L1c product. These are grids that link any OLCI pixel in the reference band to the corresponding sub-pixel location in the other OLCI and SLSTR bands such that if a detector were placed at the sub-pixel location it would have seen the same target on Earth as the reference pixel.

ADS	Annotation DataSet
CFI	Customer Furnished Item
DEM	Digital Elevation Model
ECMWF	European Centre for Medium-range Weather Forecasting
ESA	European Space Agency
FR	(OLCI) Full Resolution
GMES	Global Monitoring for Environment and Security
НКТМ	Housekeeping Telemetry
IR	Infra-Red
MDS	Measurement DataSet
N/A	Non Applicable
NRT	Near Real Time
O-GPP	Optical Ground Processor Prototype
O-SPS	Optical System Performance Simulator
OLCI	Ocean and Land Colour
OLCI	Ocean and Land Colour Instrument
OZA	Observation Zenith Angle
PDGS	Payload Ground Data/Service Segment
RR	(OLCI) Reduced Resolution
SCCDB	Satellite Characterization and Calibration Data Base
SLSTR	Sea and Land Surface Temperature
SOW	Statement Of Work
SRD	System Requirements Document
SSD	Spatial Sampling Distance
SSP	Sub-Satellite Point
SZA	Sun Zenith Angle
TOA	Top Of Atmosphere
TBC	To Be Confirmed
TBD	To Be Defined
TPS	Thin-Plate Spline
ТМ	Telemetry
URD	User Requirement Document

## 1.3.3 Acronyms



## **1.4** Overview of instruments and data acquisition baseline

OLCI and SLSTR data acquisition baseline are respectively described in [RD-1] and [RD-2]. Only the characteristics of the different OLCI and SLSTR spectral channels are recalled in Table 1-1 below.

### 1.4.1 OLCI and SLSTR channels characteristics

See Table 1-1.

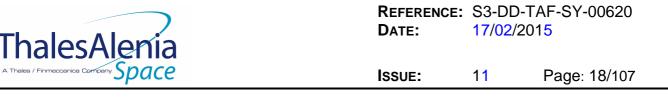


**REFERENCE:** S3-DD-TAF-SY-00620 **DATE:** 17/02/2015

**Issue:** 11 Page: 17/107

S3 channel(*)	Central Wavelength		Width		Ground resolution(**)	
	OLCI	SLSTR	OLCI	SLSTR	OLCI (nominal nadir)	SLSTR (nominal at SSP for Nadir view ; nominal constant value of backward view)
	nm	μm	nm	nm	km	km
Oa1	400		15		0.27	
Oa2	412.5		10		0.27	
Oa3	442.5		10		0.27	
Oa4	490		10		0.27	
Oa5	510		10		0.27	
S1		0.555		20		0.5 ; 0.8
Oa6	560		10		0.27	
Oa7	620		10		0.27	
S2		0.659		20		0.5 ; 0.8
Oa8	665		10		0.27	
Oa9	673.75		7.5		0.27	
Oa10	681.25		7.5		0.27	
Oa11	708.75		10		0.27	
Oa12	753.75		7.5		0.27	
Oa13	761.25		2.5		0.27	
Oa14	764.37 5		3.75		0.27	
Oa15	767.5		2.5		0.27	
	778.75		 15		0.27	
Oa10 Oa17	865		20		0.27	
S3	000	0.865	20	20	0.27	0.5 ; 0.8
		0.000		20		0.0 , 0.0
Oa18	885		10		0.27	
Oa19			10		0.27	
Oa20	940		20		0.27	
0a21			40		0.27	
S4( <sup>†</sup> )		1.375	-	15		0.5 ; 0.8
S5( <sup>†</sup> )		1.61		60		0.5 ; 0.8
S6( <sup>†</sup> )		2.25		50		0.5 ; 0.8
S7		3.74		380		1;1.6
F1(***)		3.74		380		1;1.6
S8		10.85		900		1;1.6
F2(***)		10.85		900		1;1.6
S9		12		1000		1;1.6

Table 1-1: Nominal OLCI and SLSTR channels characteristics



(\*): Oa## is the updated SRD notation for OLCI channels. In this document this is simplified to O##.

(\*\*) using an altitude of 815 km, mean of the reference S3 orbit (\*\*\*): In this document F1 is referred as S10 and F2 as S11

\*\*): In this document F1 is referred as S10 and F2 as S11

(<sup>†</sup>): Each of the SLSTR S4, S5 and S6 channels actually acquire 2 images shifted along scan and destined to be averaged at level 1b to improve SNR.

## 1.4.2 OLCI and SLSTR acquisition scheme and pixel numbering

OLCI and SLSTR acquisition scheme and pixel numbering are summarized in the two figures below.

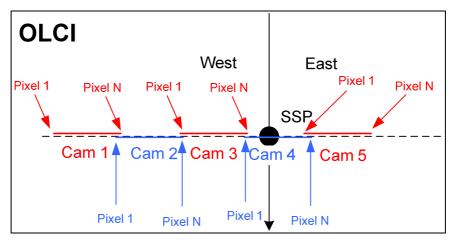


Figure 1-1: OLCI acquisition scheme and pixel numbering

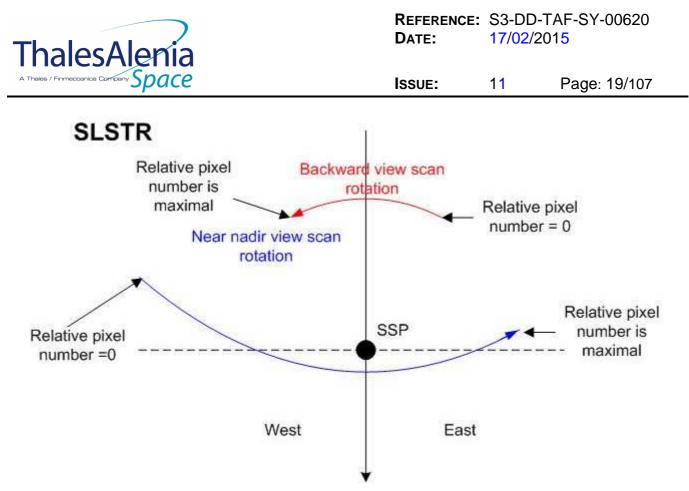


Figure 1-2: SLSTR nadir and oblique views acquisition scheme and pixel numbering. The figure represents the scan rotation direction in the current SLSTR design. Nevertheless the L1c processing can deal with the two possible rotation directions.

## 2. LEVEL 1c OVERVIEW AND BACKGROUND INFORMATION

## 2.1 Synergy Product Definition

## 2.1.1 SRD Definition

The definition of the optical end-user Level 1c product has evolved during phase B2. The latest definition in the SRD [AD03] is given below:

Synergy product

A synergy product (formerly called "Vegetation product") defines the level 1c. This product will include all\* OLCI bands and SLSTR channels from nadir and oblique views. The product is acquired continuously over ocean and land areas in full resolution. The level 1c data will include OLCI and SLSTR L1b data with no further radiometric processing. All spectral channels will be referenced to one instrument grid of a specific OLCI camera geometry. However. The L1c product is not resampled to a specific surface grid or projection but includes all the necessary misregistration information so that any user-defined projection or gridding can be performed at a higher level.

The synergy product will be annotated with parameters as specified below. In addition the annotation will include:

• Misregistration information between OLCI and SLSTR.



**Issue:** 11 Page: 20/107

This product will be delivered with two levels of timeliness:

- NRT 3 hours product, assuming 1 hour for ground processing and 2 hours for satellite acquisition and downlink.
- Archived product without any specific timeliness requirement.

With adequate atmospheric correction, the synergy product (L1c) can be transformed into Level 2 Bottom of the Atmosphere geophysical parameters.

Common annotation data for level 1 optical data products

Data products from OLCI and SLSTR at Level 1b and 1c will be annotated with the following information:

a) General annotation data:

- Ortho-rectified geolocation information (Lat, Lon, Altitude);
- Geophysical atmosphere information (e.g., ECMWF);
- Illumination and observation geometry;
- Parameters to convert from radiances to TOA reflectance;

b) Quality indicators:

- Preliminary pixel classification:
  - o static flags: land, saline water, coastline, tidal regions, fresh water rivers and lakes;;
  - o dynamic flags: clouds;
- Technical quality flags (e.g. cosmetic fill. Saturation)

\* The SLST L1b product contains 2 or 3 bands for each of the S4, S5 and S6 channels: 2 shifted sub-bands A and B acquired by the instrument and an optional "averaged" band A+B computed at level 1b (option activated through L1b user configurable parameters). It has been agreed that only one sub-band per channel is included in the L1c product. The proposed L1c processing can indifferently process any sub-band, chosen by the user as input parameter. The precise definition of contents and format of L1c product is given in [RD-5]. Only specific features of the L1c product are enumerated here:

- Level 1c product is computed:
  - from Full Resolution OLCI L1b products and SLSTR L1b products (including nadir and oblique views). Concerning SLSTR SWIR channels (S4 to S6), in baseline only the L1b "averaged" (or TDI) channels are considered at level 1c, although the L1c processing is generic enough to use the "A", "B" or TDI SWIR sub-bands.
  - on the common part of OLCI and SLSTR nadir view swaths (OLCI swath around 1250km is included in the SLSTR nadir view swath). As a consequence the areas of the SLSTR nadir and oblique views L1b images that are not covered by the OLCI image are rejected by cutting the images along-track and across-track.
  - for the daylight part of the Sentinel 3 orbit (OLCI acquisition). The daylight part of the orbit is defined as the part of the orbit where the sun zenith angle at satellite ground track is lower than 80°.
- One OLCI reference channel, noted O<sub>q</sub>, is the reference for ortho-rectified geolocation. Thus, this channel is ortho-geolocated with the (longitude, latitude, altitude) coordinates from level 1b data.
- Misregistration measurements are included into the L1c product: for each pixel (*k*,*j*) of the OLCI reference channel  $O_q$  of each camera module m (m = 1...5), the corresponding co-



**ISSUE:** 11 Page: 21/107

registered sub-pixel location in all other OLCI and SLSTR channels are given in the level 1c product.

- Most of the OLCI and SLSTR L1b annotations are reproduced in the level 1b product (see RD-5) for a summary of these annotations. When two sets of annotations are redundant, the OLCI one is chosen.
- Most of the data are gridded onto one of the 6 kind of grids described in paragraph 2.1.2.

From L1c product, pixels can be resampled onto any user defined grid, taking into account:

- Spatial co-registration between spectral bands
- OLCI swath discontinuity due to its 5 camera modules
- Terrain elevation to correct parallax distortion and to provide ortho-images

## 2.1.2 L1c product grids

Due to the storage of OLCI and SLSTR (nadir and oblique) images in their specific acquisition geometry along with their annotations, there are 5 kinds of grid to be considered in the level 1c product:

- 1 OLCI Pixel Resolution (PR) grid for each camera module image, corresponding to each OLCI camera module image m = 1 to 5 in its acquisition geometry. The size of these grids is N\_LINE\_OLC x N\_DET\_CAM
- 1 SLSTR Nadir view 500m Pixel Resolution (NPR05km) grid, corresponding to the SLSTR nadir-view image in its acquisition geometry (at the resolution of the 500m channels). The size of this grid is N\_SCAN\_SLST\_NAD\_05km\_L1C x N\_PIX\_SLST\_NAD\_05km\_L1C.
- 1 SLSTR Nadir view 1km Pixel Resolution (NPR1km) grid, corresponding to the SLSTR nadir-view image in its acquisition geometry (at the resolution of the 1km channels). The size of this grid is N\_SCAN\_SLST\_NAD\_1km\_L1C x N\_PIX\_SLST\_NAD\_1km\_L1C.
- 1 SLSTR Oblique view 500m Pixel Resolution (APR05km) grid, corresponding to the SLSTR oblique view image in its acquisition geometry (at the resolution of the 500m channels). The size of this grid is N\_SCAN\_SLST\_ALT\_05km\_L1C x N\_PIX\_SLST\_ALT\_05km\_L1C.
- 1 SLSTR Oblique view 1km Pixel Resolution (APR1km) grid, corresponding to the SLSTR oblique view image in its acquisition geometry (at the resolution of the 1km channels). The size of this grid is N\_SCAN\_SLST\_ALT\_1km\_L1C x N\_PIX\_SLST\_ALT\_1km\_L1C.

## 2.2 Objectives and outlines of Level 1c Processing

The main objective of the L1c algorithms described in this document is to deliver an accurate estimation of the misregistration between all bands (21 OLCI + 11 SLSTR bands) of OLCI FR and SLSTR nadir view images, starting from L1b products. It supports the requirement PL-OP-030 of the ESA SRD [AD03] recalled below:

The OLCI and SLSTR near nadir channels shall be co-registered over Land within 0.4 (0.3 goal) SSD (rms) of the OLCI FR SSD as specified in requirements OL-DE-130 and OL-GE-020.

Resampling is not performed at level 1c, but level 1c shall include all the necessary information to perform user-defined projection at a higher level (i.e. Level 2). For the performance budget calculation, the influence from the DEM (provided as CFI) can be considered as negligible.



Note that this requirement does not apply to the SLSTR oblique view. Nevertheless at level 1c the oblique view is collocated with the OLCI reference channel by means of ortho-rectified geolocation information.

In this context two kinds of misregistration are distinguished:

- Inter-instrument spatial misregistration: misregistration between one OLCI reference channel and one SLSTR reference channel.
- **Intra-instrument spatial misregistration**: misregistration between one reference channel and all the other channels within a same instrument.

The main task of level 1c processing is to estimate the dynamic inter-instrument spatial misregistration between two spectrally similar OLCI and SLSTR channels. The intra-instrument spatial misregistration is obtained by on-ground and/or in-flight characterization/calibration and will be handled (without enhancement) in the level 1c product.

At level 1c:

- the OLCI reference channel is noted O<sub>q</sub>, with q = 17 in baseline (see section 3.2.1.1). The OLCI reference channel is selected by the user using the L1c\_OLCI\_ref\_band processing parameter.
- the SLSTR reference channel is noted  $S_u$  with u = 3 in baseline (see section 3.2.1.1). The SLSTR reference channel is selected by the user using the  $L1c\_SLSTR\_ref\_band$  processing parameter.
- the 5 OLCI camera modules are considered separately, so the 21 O<sub>b</sub> channels (b=1,...,21) are split into 5 pieces: O<sub>b</sub><sup>m</sup>, m=1,...,5.

Figure 2-1 exhibits the general procedure executed at Level 1c for the near-nadir views. It compounds four main steps:

- 1. Pre-processing of the inputs: the main goal is to adapt the L1b data in input to the subsequent processing,
- 2. Inter-Instrument misregistration estimation: for each OLCI camera module m (m=1,...,5), it puts in correspondence the reference SLSTR Visible channel Su with the reference OLCI channel O<sub>q</sub><sup>m</sup>. This is the bulk of the L1c processing.
- Computation of correspondence between the OLCI O<sub>q</sub> channel, taken as reference for orthorectified geolocation, and all other OLCI and SLSTR channels, based on the misregistration estimated at previous step and on intra-instrument misregistration estimation stored in characterization auxiliary data files.
- 4. Creation of the L1c product annotations, mostly from L1b annotations.

Note that the correspondence with the SLSTR oblique view channels is computed at stage #3, based on OLCI and SLSTR oblique view ortho-rectified geo-location information.



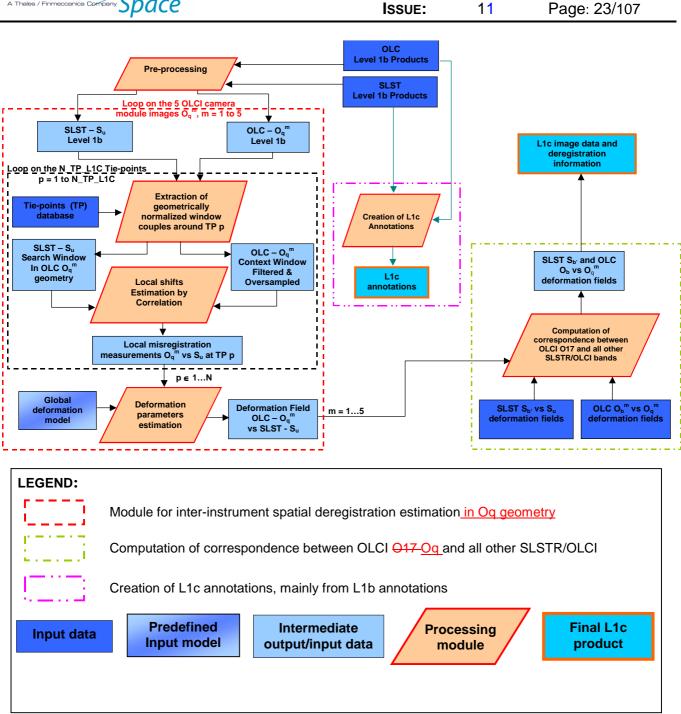


Figure 2-1: L1c Processing General Scheme

## 3. ALGORITHMS DESCRIPTION

- 3.1 Pre-processing
- 3.1.1 Algorithm Inputs



## 3.1.1.1 OLCI FR L1b product and SLSTR L1b product

These products are described in [RD-3] and [RD-4]. The OLCI FR and SLSTR L1b products must:

- cover the daylight part of the orbit
- include at least:
  - All the pixels present in the L1b processing before the geometric L1b resampling. They correspond to L0 pixels with L1b radiometric calibration applied.
  - Ortho-rectified geolocation information allowing to retrieve the ortho-rectified geolocation (lon,lat,alt) of all the acquired pixels of all channels
  - o Quality flags
  - Data necessary to convert L1b product TOA radiance channels to TOA reflectance

The two products have been acquired during the same orbit.

### 3.1.1.2 A tie points database

This auxiliary data file, described in [RD-5] includes the following information, for each tie-point:
(longitude, latitude, altitude) coordinates

• size of the context window to be extracted around the tie-point (on ground, in meters) (see section 3.2.2.1)

The number of tie-points in the database is noted N\_TP\_DB.

A tie point locates an object that is known to be "salient" when observed in a chosen spectral band by OLCI or SLSTR. Salient means that the imagette extracted around this point will have good correlation properties. Generally, this requires that the image of the located object:

- is contrasted
- is spatially limited in a small area (smaller than the correlation window)
- contains high spatial frequency components
- is not periodic
- is stable under illumination and seasonal changes

### 3.1.1.3 Processing parameters

- SLST\_SWIR\_SELECT
- TP\_SELECT\_SWITCH
- ALT\_TP\_STEP
- ACT\_TP\_STEP
- ACT\_TP\_NUM
- ALT\_TP\_NUM
- TP\_REJECTION\_TESTS\_SWITCHES
- OLC\_SEGMENT\_SIZE (section 3.1.4.6)
- W\_ACT\_TP\_MARGIN(m), m = 1 to 5
- E\_ACT\_TP\_MARGIN(m), m = 1 to 5
- ALT\_TP\_MARGIN
- SLST\_LAT\_MARGIN
- p\_ref\_OL\_TS
- p<sub>n</sub>\_ref\_SL\_NAD\_05km\_TS



**Issue:** 11 Page: 25/107

- *p<sub>n</sub>\_ref\_SL\_NAD\_1km\_TS*
- p<sub>a</sub>\_ref\_SL\_ALT\_05km\_TS
- p<sub>a</sub>\_ref\_SL\_ALT\_1km\_TS
- UNIT CONV PARAM
- Interpolation parameters
- L1c\_OLCI\_ref\_band
- L1c\_SLSTR\_ref\_band

## 3.1.2 Processing Objective

The goal of pre-processing is to retrieve and condition input data for L1c subsequent processing. It performs the following operations:

## 3.1.2.1 Retrieve the full images of the 5 OLCI camera modules in their acquisition geometry from L1b data, and their annotations

During the level 1b processing, instrument pixels have been resampled (with a nearest neighbor method) to an OLCI L1b specific grid not appropriate for L1c processing. Moreover, some pixels located in the overlapping areas between two adjacent camera modules are not used to construct the L1b image, but they are kept in a specific dataset of the L1b product (they are called "removed pixels"). Hence this step of the pre-processing stage aims at reconstructing the full images (overlapping areas included) of the five camera modules in a more basic geometry, called the OLCI "acquisition geometry" (see precise definition in section 3.1.4.2).

The L1b annotations required for the L1c processing and/or to be included in the L1c product are also retrieved from L1b products at this stage.

# 3.1.2.2 Retrieve the channels of the SLSTR nadir and oblique view in their acquisition geometry from L1b data, and their annotations

During the level 1b processing, oblique view and nadir view instrument pixels have been resampled (nearest neighbour) to a SLSTR L1b specific grid not appropriate for L1c processing. Moreover, some pixels may have been lost (see [RD-7]) during the regridding, but they are kept in the L1b product (they are called "orphan pixels"). Hence this step of the pre-processing stage aims at reconstructing the full SLSTR nadir and oblique view image in a more basic geometry, called the SLSTR "acquisition geometry" (see precise definition in section 3.1.4.3).

Coarse along-track boundaries of the SLSTR nadir and oblique view images are also computed in order to keep only the part of the SLSTR channels whose extension is covered by the OLCI images.

The L1b annotations required for the L1c processing and/or to be included in the L1c product are also retrieved from L1b products at this stage.

## 3.1.2.3 Convert the OLCI and SLSTR nadir view reference channels from TOA radiance to TOA reflectance

The radiometric unit of the OLCI and SLSTR reference channels (respectively  $O_q$  and  $S_u$ ) in the L1b products is TOA radiance.



Before performing the correlation step, the level 1c processing must convert the radiometric measurements of the processed channels into TOA reflectance, by applying conversion coefficients computed from data included in the L1b products.

## 3.1.2.4 Tie points selection

From the tie points database (paragraph 3.1.1.2), this pre-processing selects the tie-points that lie in the ground area covered by the each one of the 5 OLCI camera module image processed. Only those tie-points will be needed to process the current image at Level 1c.Tie points close to the edges of the OLCI L1b image are rejected.

## 3.1.3 Algorithm Outputs

The pre-processing stage gives several outputs, which are to be used as inputs by the subsequent L1c processing module:

- Five multiband OLCI FR images in their acquisition geometry, one for each camera module m:
  - Calibrated radiometric measurements from L1b (TOA radiance)
  - Gridded on the OLCI acquisition grid (PR grid, see sections 2.1.2 and 3.1.4.2.1)
  - With the L1b annotations defined in [RD-5];
  - With each pixel of the  $O_q^m$  reference channel ortho-geolocated.
- One multiband SLSTR nadir view image in its acquisition geometry:
  - Calibrated radiometric measurements from L1b (TOA radiance and brightness temperature)
  - Gridded on the SLSTR nadir view acquisition grid (NPR grid, see sections 2.1.2 and 3.1.4.3.1)
  - With the L1b annotations defined in [RD-5].
  - With each pixel of the S<sub>u</sub> reference channel ortho-geolocated
- One multiband SLSTR oblique view image in its acquisition geometry:
  - Calibrated radiometric measurements from L1b (TOA radiance and brightness temperature)
  - Gridded on the SLSTR oblique view acquisition grid (APR grid, see sections 2.1.2 and 3.1.4.3.1)
  - o With the L1b annotations defined in [RD-5].
  - With each pixel of the S<sub>u</sub> reference channel ortho-geolocated
- Duplicated OLCI O<sub>q</sub><sup>m</sup> and SLSTR S<sub>u</sub> reference bands converted into TOA reflectance or normalized TOA radiance (as defined by UNIT\_CONV\_PARAM), to be further processed
- $s_{\min}^{a,1km}$ ,  $s_{\min}^{a,05km}$ ,  $s_{\min}^{n,1km}$ ,  $s_{\min}^{n,05km}$  that represent the scan number of the first scan of the part of the SLSTR nadir and oblique views 500m and 1km channels kept for further processing (after a rough selection of the part of the SLSTR product that is covered by the OLCI product, see section 3.1.4.3.2.1)
- 5 subsets of the tie-points database and their corresponding location (k<sub>p</sub>, j<sub>p</sub>) in the OLCI camera module images (computed in section 3.1.4.6): one per OLCI camera module image m, to be used by the L1c processing, containing N\_TP\_L1C(m) tie points, m = 1 to 5. If the tie-points come from the tie-points database their index into the database shall also be associated with their (k<sub>p</sub>, j<sub>p</sub>) coordinates.



**ISSUE:** 11 Page: 27/107

All the above listed data shall be output as intermediate verification data by the Processor.

## 3.1.4 Mathematical Description

#### 3.1.4.1 Check inputs

Before going into further algorithmic processing, the processor shall verify the consistency of <u>all</u> its input parameters with authorized range or possible values defined in SY-4. In particular:

- If L1c\_SLSTR\_ref\_band is between 4 and 6 (SWIR channel) and corresponding SLST\_SWIR\_SELECT.S{L1c\_SLSTR\_ref\_band} ≠ "A" then warn the user of the insconsistency of the parameters and stop processing. (the selected SWIR sub-band must be "A" if the reference channel is a SWIR band).
- If *TP\_SELECT\_SWITCH* == "REGULAR\_STEP" OR SELECT\_SWITCH == "REGULAR\_N" then check that input parameter *CW\_SIZE\_SWITCH* == "FIXED" (Parameter to be used in section 3.2). If it is not thes case, warn the user and stop processing
- If CW\_SIZE\_SWITCH=="FIXED" then check that CW\_K\_RADIUS is set with a value. If it is not the case warn the user and stop processing.
- If E\_ACT\_TP\_MARGIN (or W\_ACT\_TP\_MARGIN or ALT\_TP\_MARGIN) < CW\_K\_RADIUS + 15 then warn the user: tie-points close to the edges of the image may be rejected due to insufficient margins. Continue processing.
- If CW\_SIZE\_SWITCH == "FIXED" and T\_SIZE\_CW < CW\_K\_RADIUS then warn the user that all the tie points will be rejected because T\_SIZE\_CW < CW\_K\_RADIUS, but continue processing.
- Check that SLST\_1km\_K\_MARGIN and SLST\_1km\_J\_MARGIN are even number. If not, stop processing.

3.1.4.2 Retrieve the full images of the 5 OLCI camera modules in their acquisition geometry from L1b data, and their annotations

3.1.4.2.1 Definition of OLCI "acquisition geometry"

Each camera module m of OLCI acquires a complete row of N\_DET\_CAM spatial pixels (including 21 spectral values, indexed by b) at regularly spaced instants  $t_k = t_0 + k.T_s$ ,  $k = 0,...,N\_LINE\_OLC-1$ , where  $T_s$  is the time sampling interval. The row of the first acquired spatial pixels included in the L1b FR product is indexed by k=0. The row of the last acquired spatial pixels included in the L1b FR product is indexed by k=N\_LINE\_OLC-1. Spatial pixels in a same row are indexed with j=0...N\_DET\_CAM-1 (columns of the image).

Thus each radiometric value in the L1b OLCI FR product can be uniquely indexed with b,m,k,j. The image of the OLCI camera module *m* in its acquisition geometry is defined as:

 $I_b^m(k,j)$  eq. 3-1



**ISSUE:** 11 Page: 28/107

with  $k = 0...N_LINE_OLC-1$  and  $j = 0...N_DET_CAM-1$  respectively indexing the rows and columns of the image; b = 1...21 is the spectral channel; m = 1...5 is the camera module number. N\_DET\_CAM is the total number of spatial detector cells used for acquisition in camera module m. N\_LINE\_OLC is the total number of acquired frames in the L1b product. It can vary within a small interval from one product to another.

Remark: N\_DET\_CAM is the same for all camera module number m (N\_DET\_CAM = 740 from current OLCI design).

3.1.4.2.2 Retrieving the 5 camera module images in their acquisition geometry and per pixel annotations

This processing mainly *rearrange* the pixels from the L1b product, including the so-called removed (or ungridded) pixels appended to the L1b product, to the 5 L1c Pixel Resolution (PR) grids of each camera module image (see description of the L1c grids in paragraph 2.1). This is done using information attached to each L1b pixel (its band, detector index product frame number and frame offset). At the same time, the L1b per pixel (i.e. on the L1b FR product grid) annotations are retrieved and attached to each pixel.

The OLCI per pixel annotations included in the level 1b product that must be handled by the Level 1c processing are the following ones (extracted from [RD-4]):

- Ortho-rectified geolocation at pixel level,
- Quality flags according to the list provided in the L1c product definition [RD-5].
- The following annotations are also retrieved from L1b annotations and processed:
  - The time-stamps of each line of the OLCI image
  - The subsampled Solar Azimuth angle (SZA) annotations are extended to each pixel (used to convert the reference channel from radiance to reflectance unit)

The algorithm is as follows:

Create the 5 x 21 OLCI image structures  $I_b^m(k, j)$  (see eq. 3-1) for m=1 to 5, b=1 to 21, k=0 to N\_LINE\_OLC-1, j=0 to N\_DET\_CAM-1, and set each element to PIXEL\_UNFILLED value.

Read the frame offset array in the OLC General information data file.

Find the minimum frame\_offset min\_ $\delta$ f over all the frame offset

For each gridded spatial pixel  $(f, j_{L1b})$  of the OLCI L1b product

*II* Find the corresponding camera module m and location (k,j) in camera module m: Retrieve the detector index p of the product pixel  $(f, j_{L1b})$  from the L1b product annotations (detector\_Index field in the General Information data file)

Retrieve the frame offset  $\delta f$  corresponding to detector p from the L1b product annotations (Frame\_offset field in the General information data file)

Retrieve the camera module m that acquired pixel (f,j<sub>L1b</sub>):  $\mathbf{m} = FLOOR[p/N_DET_CAM] + 1$ 

Retrieve the across-track pixel index j in the camera module image m:  $\mathbf{j} = p - (m-1)^*$ N\_DET\_CAM

Retrieve the instrument frame index k:  $\mathbf{k} = f - \delta f + \min_{\delta} f$ // Regrid radiometry:



Set  $I_b^m(k, j)$  to the radiometric values (for all bands b) associated to spatial pixel (f,j<sub>L1b</sub>) of the OLCI L1b product (TOA\_Radiances field in the TOA Radiances band ADS in the Radiances files)

The error\_estimates parameter shall also be regridded

## // Regrid L1b per pixel annotations:

Read per pixel annotations associated to L1b product pixel (f,j<sub>L1b</sub>). It concerns:

- Ortho-rectified geolocation: longitude, latitude and altitude corrected from DEM. Stored in the Geolocation file of the OLCI L1b product
- Quality flags, stored in the Quality flags ADS of L1b product according to the list provided in the L1c definition [RD-5]. Note: The "Duplicated Pixel" flag become a spare bit at L1c

Attach these annotations to the L1c pixel (k,j) of camera module image m

## // Retrieve per pixel Sun Zenith angle (SZA) and regrid:

Compute the SZA for L1b product pixel  $(f, j_{L1b})$  by linear interpolation of the SZA given at tie-points in the Tie Point Annotations file of L1b product (see [RD-4]).

Attach the value to location (k,j) in the SZA grid of camera module m, noted  $\theta_{SZA,OLCI}^{m}(k,j)$ .

## // Obtain L1c OLCI time-stamps from L1b ones:

If  $p = p\_ref\_OL\_TS$  then read the f<sup>th</sup> element *ts(f)* of the L1b time stamp dataset and assign *ts(f)* +  $\delta f^* \Delta T\_OLCI$  to the k<sup>th</sup> element of the L1c OLCI time stamps dataset.  $p\_ref\_OL\_TS$  is the index number of the reference (SSP pointing) OLCI pixel;  $\Delta T\_OLCI$  is the time interval between two frames acquired by OLCI.

End For

For each removed spatial pixel of the OLCI L1b product (in the Removed Pixels file) (its product frame is noted f)

// Find the corresponding camera module m and location (k,j) in camera module m: Retrieve the detector index p of the removed pixel from the L1b product annotations (RP\_detector\_index field in the Removed Pixels file)

Retrieve the camera module m that acquired the removed pixel:  $\mathbf{m} = FLOOR[p/N_DET_CAM] + 1$ 

Retrieve the across-track pixel index j in the camera module image m:  $\mathbf{j} = p - (m-1)^* N_DET_CAM$ 

Retrieve the instrument frame index k:  $\mathbf{k} = f + \min_{\delta} f$ // Regrid radiometry:

Set  $I_b^m(k, j)$  to the radiometric values (for all band b) associated to the removed spatial pixel of the OLCI L1b product (21 RP\_TOA\_Radiances variables in the Removed Pixels file)

## // Regrid L1b per pixel annotations:

Read per pixel annotations associated to L1b removed pixel. It concerns:

- Ortho-rectified geolocation: longitude, latitude and altitude corrected from DEM. Stored in the RP\_longitude, RP\_latitude, RP\_altitude fields
- Quality flags, stored in the RP\_quality\_flags field according to the list provided in the L1C definition [RD-5]. Note: The "Duplicated Pixel" flag become a spare bit at L1c

Attach these annotations to the L1c pixel (k,j) of camera module image m



**ISSUE:** 11 Page: 30/107

## // Retrieve per pixel Sun Zenith angle (SZA) and regrid:

Read the value (RP\_SZA) corresponding to the processed removed pixel and attach the value to location (k,j) in the SZA grid of camera module m, to complete the  $\theta_{SZA,OLCI}^{m}(k, j)$  grid.

#### End For

At the end of this regridding process all the elements of the  $I_b^m(k, j)$  arrays must have been filled (no PIXEL\_UNFILLED value must remain).

Note: The OLCI  $O_q$  channel is ortho-geolocated with the information included per pixel in the OLCI L1b product. There is one ortho-rectified geolocation grid per camera module image. The 5 ortho-rectified geolocation grids are given by the DEM corrected (longitude, latitude) data indexed by the pixel coordinates of the channel  $O_q^m$  in its acquisition geometry.

3.1.4.2.3 Retrieving sub-sampled and general OLCI L1b annotations

## 3.1.4.2.3.1 OLCI L1b sub-sampled and general Annotations

The OLCI sub-sampled and general annotations (i.e. not the per pixel annotation on the L1b FR product grid) included in the level 1b product that must be handled at Level 1c are the following ones (extracted from [RD-4]):

- One Tie points (not to be confused with the L1c tie points) Annotations datafile covering geolocation, Sun and Viewing geometry, and Environment data (e.g. meteo annotations) according to [RD-4]:
  - Geolocation field shall include:
    - Longitude,
    - Geodetic latitude,
    - Altitude.
  - Sun and Viewing Geometry field shall include:
    - Sun Zenith Angle,
    - Sun Azimuth Angle,
    - Viewing Zenith Angle,
    - Viewing Azimuth Angle.
  - Meteo annotations field (contents given in [RD-4])
- One General Information Data File including Instrument Data that are useful for the use of the products.

### 3.1.4.2.3.2 Annotations Processing

3.1.4.2.3.2.1 Processing of the annotations in the Tie points Annotations datafile

Concerned annotations are:

- Geolocation, Sun and Viewing Geometry annotations
- Meteo annotations



**Issue:** 11 Page: 31/107

The processing consists in gathering for each L1b tie-point the geolocation (longitude, latitude, altitude), the corresponding Sun geometry (SZA, SAA), viewing geometry (OZA, OAA) and meteo L1b annotations, to be provided in the L1c product [RD-5].

### 3.1.4.2.3.2.2 Additional Data Sets

The process consists in copying the OLC product General Information data file after removing the following variables:

- detector\_index
- frame offset

3.1.4.3 Retrieve the SLSTR channels in their acquisition geometry from L1b data, and related per pixel annotations

3.1.4.3.1 Definition of SLSTR "acquisition geometry"

Each SLSTR measurement of the nadir or oblique view can be indexed with k', j', b', respectively the scan trace number, the relative pixel number of the scan trace k' and the spectral channel.

Here *k*' identifies one scan "trace", being understood that several scan "traces" are acquired at each physical scan S due to SLSTR design:

- 4 scan lines for the VIS (S1-S3) channels and for any sub-band (A, B or TDI) of the SWIR (S4-S6) channels (considering that an averaging processing at Level 1b has reduced the number of logical elements from 8 to 4 in SWIR TDI sub-bands)
- 2 scan lines for the MWIR (S7), TIR (S8-S9) and F1 & F2 channels.

The SLSTR channel b' in its acquisition geometry is defined as:

 $I_{b'}^{view}(k', j')$ 

with *view* = nadir or oblique, k' indexing the lines of the image and j' indexing the columns of the nadir view or oblique view image.

See [RD-3] for a definition of the terms scan number *S*, scan trace number, instrument absolute index, instrument relative index and detector index.

 $k' = 0...N_SCAN_SLST_NAD_1km - 1$  (resp. N\_SCAN\_SLST\_NAD\_05km - 1) for the nadir view 1 km (resp. 500m) channels or  $k' = 0...N_SCAN_SLST_ALT_1km - 1$  (resp. N\_SCAN\_SLST\_ALT\_05km - 1) for the oblique 1km (resp. 500m) channels.

j' = 0...N\_PIX\_SCAN\_NAD\_1km - 1 (resp. N\_PIX\_SCAN\_NAD\_05km - 1) for nadir view 1km (resp. 500m) channels or j' = 0...N\_PIX\_SCAN\_ALT\_1km -1 (resp. N\_PIX\_SCAN\_ALT\_05km -1) for oblique view 1km (resp. 500m) channels.

N\_PIX\_SCAN\_NAD\_##km and N\_PIX\_SCAN\_ALT\_##km are constant numbers, independent of k'. N\_SCAN\_SLST\_\$\$\$\_##km can vary within a small interval from one product to another.

Notes:

• For nadir or oblique view, the relation between the scan number S, the detector number  $k_{det}$  and the scan trace number k' is: k' = 2.scale.S +  $k_{det}$ . With  $k_{det}$  = 0 or 1 for the



**ISSUE:** 11 Page: 32/107

thermal/1km channels,  $k_{det} = 0,...,3$  for the solar/500m channels. *scale* is 1 for the thermal channels and 2 for the solar channels.

• In L1b products, there is one more column j in TDI channels than in the VIS and SWIR A/B channels. At level 1c, the pixels of the lastest column and their annotations of the TDI channels are not kept and all the 500m channels are supposed to have the same number of columns.

3.1.4.3.2 Retrieving the SLSTR nadir and oblique view images in their acquisition geometry and per pixel annotations

The processing mainly *rearrange* the pixels from the L1b product, including orphan (or ungridded) pixels appended to the L1b product, to the L1c SLSTR Pixel resolution grids (NPR##km and APR##km, see description of the L1c grids in paragraph 2.1). This is done using information attached to each L1b pixel (its band, scan line number, pixel number on the scan line). The images are also cut in order to keep only the scans that have been acquired during the OLCI acquisition (plus a margin). At the same time, the L1b nadir and oblique view per pixel annotations are retrieved and attached to each pixel.

The SLSTR per pixel annotations included in the level 1b product that must be handled by the Level 1c processing are the following ones (extracted from [RD-3]):

- Full resolution geodetic coordinates Data files: These files include the ortho-rectified geolocation of all instrument pixels (latitude, longitude and altitude) for each sub-bands ("A", "B", TDI and 1km) of the nadir and oblique views. In practice only the "A stripe" orthorectified geolocation grid is required for nadir view Su channel, while 2 ("A stripe "and 1km detectors) to 4 ("A stripe ", "B stripe", TDI detectors and 1km detectors) grids are required for the oblique view depending on the selected sub-bands for SWIR channels.
- Quality flags for nadir and oblique views according to the list provided in the L1c definition [RD-5].
- Scan, pixel and detector number files for 500m and 1km pixels, nadir view image
- Scan, pixel and detector number files for 500m and 1km pixels, oblique view image

The following annotations are also retrieved from L1b annotations and processed:

- The time-stamps of each k' at 500m and 1km resolution for the both views are also retrieved from L1b time information.
- The subsampled Solar Azimuth angle (SZA) annotations are extended to each 500m pixel of the nadir view (used to convert the reference channel from radiance to reflectance unit, see section 3.1.4.4)

The algorithms for nadir and oblique views are described below.

From now on, only the SLSTR SWIR sub-bands (included in the L1b product) selected in the input parameter SLST\_SWIR\_SELECT are considered as the L1c SWIR channels (for nadir and oblique views):

If SLST\_SWIR\_SELECT.S# = "A" then consider the "A" sub-band (or "A stripe") in channel S# at level 1c

If SLST\_SWIR\_SELECT.S# = "B" then consider the "B" sub-band (or "B stripe") in channel S# at level 1c

If SLST\_SWIR\_SELECT.S# = "A+B" then



**ISSUE:** 11 Page: 33/107

- Check the presence of TDI channels in the L1b product by reading the value of the *Tdi\_switch* parameter in the SLST L1b SPH.
- If there the TDI channels have not been processed at L1b then warn the user with the inconsistency between the current *SLST\_SWIR\_SELECT* parameter and stop the processing.
- Else consider the "averaged" (or TDI) sub-band in channel S# at level 1c (this is the baseline for all SWIR channels).

# is to be replaced by 4, 5 and 6. In the rest of the document, whatever the selected sub-band, the SWIR channels are noted S4, S5 and S6.

Note that the S1, S2 and S3 channels are considered as "A sub-bands".

3.1.4.3.2.1 Finding coarse oblique limits of the SLSTR images

This first step finds the instrument scans numbers  $s_{\min}^{n,lkm}$ ,  $s_{\max}^{n,lkm}$  (resp.  $s_{\min}^{a,lkm}$ ,  $s_{\max}^{a,lkm}$ ) respectively the 1<sup>st</sup> and last instrument scans of the nadir view (resp. the oblique view) 1km channels to be considered by the L1c processing. They roughly define the part of the SLSTR 1km channels that are also covered by the OLCI image. The algorithm is described below.

Find the minimum and maximum latitude over all the pixels of the 5 OLCI ortho-rectified geolocation grids retrieved in section 3.1.4.2.2. Only the first and last lines (k = 0 and k = N\_LINE\_OLC-1) of each grid should be explored. The minimum and maximum latitude are respectively noted  $\lambda_{min}$  and  $\lambda_{max}$ .

### For nadir view:

Find the latitude of the satellite trace pointing pixel of each instrument scan in the SLSTR nadir view *L1b* image (1km channels):

In the nadir view 1km channel *Full resolution geodetic coordinates ADS*, the 3D array of variable latitude is noted  $\lambda(S^{nad},p_n,k_{det})$  where  $S^{nad}$  is the nadir view scan number,  $p_n$  is the absolute pixel number along-scan and  $k_{det}$  the detector index.

Note: the absolute pixel number  $p_n$  is obtained adding the add\_offset attribute to the pixel variable read in the ADS

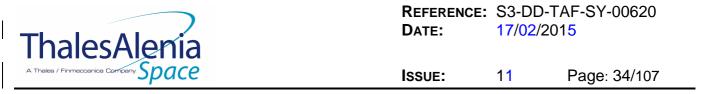
For each (S<sup>nad</sup>, k<sub>det</sub>) read  $\lambda$ (S<sup>nad</sup>, p<sub>n</sub>\_ref\_SL\_NAD\_1km\_TS, k<sub>det</sub>), now noted  $\lambda$ (S<sup>nad</sup>, k<sub>det</sub>) In the  $\lambda$ (S<sup>nad</sup>, k<sub>det</sub>) array find the element whose latitude is closest to  $\lambda_{min} - SLST\_LAT\_MARGIN$  (resp.  $\lambda_{max} + SLST\_LAT\_MARGIN$ ) and retrieve its scan number  $S_{max}^{nad}$  (resp.  $S_{min}^{nad}$ )<sup>1</sup>. The bounding instrument scan number is  $s_{max}^{n,1km} = 2(S_{max}^{nad} + 1)$  (resp.  $s_{min}^{n,1km} = 2(S_{min}^{nad} - 1)$ ).

Note N\_SCAN\_SLST\_NAD\_1km\_CUT =  $s_{max}^{n,1km} - s_{min}^{n,1km} + 1$  the number of instrument scans in the nadir view 1km channels to be output by the pre-processing stage.

The scan limits for the nadir view 500m channels are deduced from the 1km ones:  $s_{\min}^{n,05km} = 2.s_{\min}^{n,1km}$  and  $s_{\max}^{n,05km} = 2.s_{\max}^{n,1km}$ . Note N\_SCAN\_SLST\_NAD\_05km\_CUT =  $s_{\max}^{n,05km} - s_{\min}^{n,05km} + 1$  the number of instrument scans in the nadir view 500m channels to be output by the preprocessing stage.

## For oblique view:

 $<sup>^{1}</sup>$   $S_{max}^{nad}$  (resp.  $S_{min}^{nad}$ ) corresponds to  $\lambda_{min}$  (resp.  $\lambda_{max}$ ) since the orbit is descending during the daylight



Find the latitude of the satellite trace pointing pixel of each instrument scan in the SLSTR oblique view *L1b* image (1km channels).

In the oblique view 1km channel *Full resolution geodetic coordinates ADS*, The 3D array of variable latitude is noted  $\lambda(S^{alt},p_n,k_{det})$  where  $S^{alt}$  is the oblique view scan number,  $p_a$  is the absolute pixel number along-scan and  $k_{det}$  the detector index.

Note: the absolute pixel number p<sub>a</sub> is obtained adding the add\_*offset attribute* to the *pixel* variable read in the ADS

For each ( $S^{alt}$ ,  $k_{det}$ ) read  $\lambda$ ( $S^{alt}$ ,  $p_a\_ref\_SL\_ALT\_1km\_TS$ ,  $k_{det}$ ), now noted  $\lambda$ ( $S^{alt}$ ,  $k_{det}$ )

In the  $\lambda(S^{alt}, k_{det})$  array find the element whose latitude is closest to  $\lambda_{min} - SLST\_LAT\_MARGIN$  (resp.  $\lambda_{max} + SLST\_LAT\_MARGIN$ ) and retrieve its scan number  $S^{alt}_{max}$  (resp.  $S^{alt}_{min}$ ). The bounding instrument scan number is  $s^{a,1km}_{max} = 2(S^{alt}_{max} + 1)$  (resp.  $s^{a,1km}_{max} = 2(S^{alt}_{min} + 1)$ ).

Note N\_SCAN\_SLST\_ALT\_1km\_CUT =  $s_{max}^{a,1km} - s_{min}^{a,1km} + 1$  the number of instrument scans in the oblique view 1km channels to be output by the pre-processing stage.

The scan limits for the oblique view 500m channels are deduced from the 1km ones:  $s_{\min}^{a,05km} = 2.s_{\min}^{a,1km}$  and  $s_{\max}^{a,05km} = 2.s_{\max}^{a,1km}$ . Note N\_SCAN\_SLST\_ALT\_05km\_CUT =  $s_{\max}^{a,05km} - s_{\min}^{a,05km} + 1$  the number of instrument scans in the oblique view 500m channels to be output by the preprocessing stage.

3.1.4.3.2.2 Pre-Processing for the nadir view

3.1.4.3.2.2.1 Pre-Processing for 500m channels:

Create the L1c image structure including the 6 SLSTR nadir view 500m channels  $I_{b'}^{nadir}(k', j')$  for b'=1 to 6, k'=0 to N\_SCAN\_SLST\_NAD\_05km\_CUT -1, j'=0 to N\_PIX\_SCAN\_NAD\_05km-1, and set each element to PIXEL\_UNFILLED value. N\_PIX\_SCAN\_NAD\_05km is read in the SLSTR L1b product, for instance in the *Geolocation ADS* of a nadir view 500m channel it is the *n\_pixel* dimension.

## For each channel b'=1 to 6

For each spatial pixel q indexed by  $(i_{L1b}, j_{L1b})^2$  in the Visible and shortwave infrared MDS of the SLSTR nadir view b' channel in the L1b product, including orphan pixels (indexed by  $k_{L1b}$ ) do

Read *exception* byte (or *exception\_orphan* byte) of current pixel in the *Visible and shortwave infrared* MDS of channel b'.

If *unfilled\_pixel* flag is set then skip to next pixel

Read *confidence* word (or *confidence\_orphan* word) of current pixel in the *Global flags* ADS for nadir view corresponding to the sub-band of channel b' ("A stripe", "B stripe" or TDI)

If cosmetic flag is set then skip to next pixel

// Note: The pixels in the latest column of the TDI channels (b' = 4 to 6, if selected by SLST\_SWIR\_SELECT) must not be processed

// Find the corresponding location (k',j') in acquisition geometry:

<sup>&</sup>lt;sup>2</sup> 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) according to SLSTR ATBD [RD-2], whatever the convention adopted in SLSTR SY-4 [RD-3].



**ISSUE:** 11 Page: 35/107

Read the (relative) scan number S and detector number  $k_{det}$  of pixel q from the L1b product nadir view *Scan, pixel and detector number* ADS, corresponding to the sub-band of channel b' ("A stripe", "B stripe" or TDI)

Compute the corresponding instrument scan number of the pixel:  $s = 4.S + k_{det}$ If  $s > s_{max}^{n,05km}$  or  $s < s_{min}^{n,05km}$  then pass directly to the next spatial pixel in the loop

Read the (relative) *pixel* number p<sub>n</sub>' of the nadir view spatial pixel q from the suited L1b product *Scan, pixel and detector number* ADS

Note: the formerly computed absolute pixel number  $p_n$  could be obtained as:  $p_n = p_n' + 2*FIRST_NADIR_1km_PIXEL_NUMBER$ .

// FIRST\_NADIR\_1km\_PIXEL\_NUMBER is the absolute pixel number of the first 1km pixel in the nadir scans (see [RD-2]), stored in the *add\_offset* attribute of the *pixel* variable

 $\mathbf{k'} = s - s_{\min}^{n,05\,km}$ 

$$i' = p_n'$$

If b'= L1c\_SLSTR\_ref\_band Then

If current pixel is not an orphan pixel then retain the  $(k',j') \rightarrow (i_{L1b}, j_{L1b})$  correspondence for later oblique view collocation processing:

 $i_L1b_Sref(k',j') = i_{L1b}$ 

$$j_L1b_Sref(k',j') = j_{L1b}$$

Else //(current pixel is an orphan)

i\_L1b\_Sref(k',j') = NO\_CORRESP\_VAL

j\_L1b\_Sref(k',j') = NO\_CORRESP\_VAL

End If

End If

# // Regrid radiometry and exception flags:

Read the *radiance* value associated to pixel q in the nadir view *Visible and shortwave infrared* MDS corresponding to the b' channel

Set  $I_{b'}^{nadir}(k', j')$  to this radiance value

Read *exception* flag associated to spatial pixel q in the nadir view *Visible and shortwave infrared* MDS corresponding to the b' channel and attach it to the regridded pixel  $I_{b'}^{nadir}(k', j')$ 

# // Regrid per pixel ortho-rectified geolocation information of a "A" sub-band

// Note: this step is to be performed only once for one of the "A" sub-band, and then skipped in the other iterations

Read the *latitude* and *longitude* fields associated to pixel q (or equivalently S,  $p_n$ ,  $k_{det}$ ) in the *Full resolution geodetic coordinates ADS* of a nadir view "A" sub-band Store it in an appropriate structure for nadir view ortho-geolocation data

#### // Regrid L1b clouds and confidence flags:

// Note: this step gives strictly the same result for any channel acquired with the same sub-band ("A stripe", "B stripe" or TDI). Hence this step must be skipped if it has already been performed for a channel acquired with the same sub-band than channel b'

In the SLSTR L1b *Global flags* ADS for nadir view corresponding to the sub-band of channel b' ("A stripe", "B stripe" or TDI) read:

- the cloud flags of spatial pixel q
- the confidence flags of spatial pixel q



• the *pointing* flags of spatial pixel q

Store them in an appropriate array (for A, B or TDI sub-band) at location (k',j') (associated to  $I_{k'}^{nadir}(k',j')$ )

// Obtaining L1c SLSTR nadir view 500m time-stamps from L1b ones:

// Note: this step gives strictly the same result for any channel acquired with the same sub-band ("A stripe", "B stripe" or TDI). Hence this step must be skipped if it has already been performed for a channel acquired with the same sub-band than channel b'

If  $p_n = p_n\_ref\_SL\_NAD\_05km\_TS$  then

// Note:  $p_n\_ref\_SL\_NAD\_05km\_TS$  is the absolute pixel number of a reference SLSTR nadir view 500m pixel (approximately pointing to the satellite trace).

In the SLSTR L1b *Time* ADS for nadir view corresponding to the sub-band of channel b' ("A stripe", "B stripe" or TDI) read:

- the *scan\_time* element corresponding to scan number S
- the *pixel\_time* element corresponding to pixel p<sub>n</sub>'

Add the two values to obtain the time-stamp of line k' and assign it to the k'<sup>th</sup> element of an array containing the SLSTR nadir view time stamps of the corresponding sub-band (A, B or TDI).

End If

End For // (loop on q) End For // (loop on b')

3.1.4.3.2.2.2 Pre-Processing for 1km channels:

Create the L1c image structure including the 5 SLSTR nadir view 1km channels  $I_{b'}^{nadir}(k', j')$  for b'=7 to 11, k'=0 to N\_SCAN\_SLST\_NAD\_1km\_CUT -1, j'=0 to N\_PIX\_SCAN\_NAD\_1km-1, and set each element to PIXEL\_UNFILLED value. N\_PIX\_SCAN\_NAD\_1km is read in the SLSTR L1b product, for instance in the *Geolocation ADS* of a nadir view 1km channel it is the *n\_pixel* dimension.

For each channel b'=7 to 11 do

For each 1km spatial pixel q of the SLSTR nadir view in the L1b product, including orphan pixels do

Read *exception* byte (or *exception\_orphan* byte) of current pixel in the *Visible and shortwave infrared* MDS of channel b'.

If *unfilled\_pixel* flag is set then skip to next pixel

Read *confidence* word (or *confidence\_orphan* word) of current pixel in the *Global flags* for nadir view 1km resolution

If cosmetic flag is set then skip to next pixel

# // Find the corresponding location (k',j') acquisition geometry:

Read the scan number S and detector number  $k_{det}$  of the nadir view spatial pixel q from the L1b product *Scan, pixel and detector number* ADS, *nadir view, 1km resolution* 

Compute the corresponding instrument scan number of the pixel:  $s = 2.S + k_{det}$ If  $s > s_{max}^{n,1km}$  or  $s < s_{min}^{n,1km}$  then pass directly to the next spatial pixel in the loop



**ISSUE:** 11 Page: 37/107

Read the (relative) *pixel* number p<sub>n</sub>' of the nadir view spatial pixel q from the L1b product annotations (in the *Scan, pixel and detector number* ADS for nadir view 1km resolution)

Note: the formerly computed absolute pixel number  $p_n$  could be obtained as  $p_n = p_n' + FIRST_NADIR_1km_PIXEL_NUMBER$ .

FIRST\_NADIR\_1km\_PIXEL\_NUMBER is the absolute pixel number of the first 1km pixel in the nadir scans (see [RD-2]) ]), stored in the *add\_offset* attribute of the *pixel* variable

 $\mathbf{k'} = s - s_{\min}^{n,1km}$ 

 $i' = p_n'$ 

# // Regrid radiometry and exception flags:

For all b'=7 to 11, read the brightness temperature (BT) value associated to pixel q in the nadir view *Thermal infrared* MDS corresponding to the b' channel and set

 $I_{b'}^{nadir}(k', j')$  to this value

For all b'=7 to 11, read the *exception* flag associated to pixel q in the nadir view *Thermal infrared* MDS corresponding to the b' channel and attach it to the regridded pixel  $I_{b'}^{nadir}(k', j')$ 

*II* Regrid per pixel ortho-rectified geolocation information of a 1km sub-band // Note: this step is to be performed only once for one of the 1km channel, and then skipped in the other iterations

Read the *latitude* and *longitude* fields associated to pixel q (or equivalently S,  $p_n$ ,  $k_{det}$ ) in the *Full resolution geodetic coordinates ADS* of a nadir view 1km channel Store it in an appropriate structure for nadir view ortho-geolocation data *//* **Regrid L1b clouds and confidence flags:** 

In the SLSTR L1b Global flags ADS for nadir view 1km resolution read:

- the cloud flags of spatial pixel q
- the confidence flags of spatial pixel q
- the pointing flags of spatial pixel q

Store them in an appropriate array at location (k',j') (associated to  $I_{b'}^{nadir}(k',j')$ )

// Obtaining L1c SLSTR nadir view 1km time-stamps from L1b ones: If  $p_n = p_n\_ref\_SL\_NAD\_1km\_TS$  then

// Note:  $p_n\_ref\_SL\_NAD\_1km\_TS$  is the absolute pixel number of a reference SLSTR nadir view 1km pixel (approximately pointing to the satellite trace).

In the SLSTR L1b *Time* ADS for nadir view 1km resolution read:

- the scan\_time element corresponding to scan number S
- the *pixel\_time* element corresponding to pixel p<sub>n</sub>'

Add the two values to obtain the time-stamp of line k' and assign it to the k'<sup>th</sup> element of an array containing the SLSTR nadir view 1km resolution time stamps.

End If End For // (loop on q) End For // (loop on b')

At the end of this regridding process all the elements of the  $I_{b'}^{nadir}(k', j')$  arrays must have been filled (no PIXEL\_UNFILLED value must remains).



3.1.4.3.2.3 Pre-Processing for the oblique view

3.1.4.3.2.3.1 Pre-Processing for 500m channels:

Create the L1c image structure including the 6 SLSTR oblique view 500m channels  $I_{b'}^{oblique}(k', j')$ 

for b'=1 to 6, k'=0 to N\_SCAN\_SLST\_ALT\_05km\_CUT -1, j'=0 to N\_PIX\_SCAN\_ALT\_05km - 1, and set each element to PIXEL\_UNFILLED value. N\_PIX\_SCAN\_ALT\_05km is read in the SLSTR L1b product, for instance in the *Geolocation ADS* of an oblique view 500m channel it is the *n\_pixel* dimension.

#### For each channel b'=1 to 6

For each spatial pixel q indexed by  $(i_{L1b}, j_{L1b})^3$  in the *Visible and shortwave infrared* MDS of the SLSTR oblique view b' channel in the L1b product, including orphan pixels (indexed by  $k_{L1b}$ ) do

Read *exception* byte (or *exception\_orphan* byte) of current pixel in the *Visible and shortwave infrared* MDS of channel b'.

If *unfilled\_pixel* flag is set then skip to next pixel

Read *confidence* word (or *confidence\_orphan* word) of current pixel in the *Global flags* ADS for nadir view corresponding to the sub-band of channel b' ("A stripe", "B stripe" or TDI)

If *cosmetic* flag is set then skip to next pixel

// Note: The pixels in the latest column of the TDI channels (b' = 4 to 6, if selected by SLST\_SWIR\_SELECT) must not be processed

# // Find the corresponding location (k',j') in acquisition geometry:

Read the scan number S and detector number  $k_{det}$  of pixel q from the L1b product oblique view *Scan, pixel and detector number* ADS, corresponding to the subband of channel b' ("A stripe", "B stripe" or TDI)

Compute the corresponding instrument scan number of the pixel:  $s = 4.S + k_{det}$ If  $s > s_{max}^{a,05km}$  or  $s < s_{min}^{a,05km}$  then pass directly to the next spatial pixel in the loop

Read the (relative) *pixel* number  $p_n$ ' of the oblique view spatial pixel q from the suited L1b product *Scan, pixel and detector number* ADS

Note: the formerly computed absolute pixel number  $p_n$  could be obtained as:  $p_a = p_a' + 2*FIRST_ALONG_TRACK_1km_PIXEL_NUMBER$ .

// FIRST\_ALONG\_TRACK\_1km\_PIXEL\_NUMBER is the absolute pixel number of the first 1km pixel in the oblique scans (see [RD-2]), stored in the *add\_offset* attribute of the *pixel* variable

 $\mathbf{k'} = s - s_{\min}^{a,05km}$ 

j' = p<sub>a</sub>'

#### // Regrid radiometry and exception flags:

Read the *radiance* value associated to pixel q in the oblique view *Visible and shortwave infrared* MDS corresponding to the b' channel

Set  $I_{b'}^{oblique}(k', j')$  to this radiance value

<sup>&</sup>lt;sup>3</sup> 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) according to SLSTR ATBD [RD-2], whatever the convention adopted in SLSTR SY-4 [RD-3].



**Issue:** 11 Page: 39/107

Read *exception* flag associated to spatial pixel q in the oblique view *Visible and shortwave infrared* MDS corresponding to the b' channel and attach it to the regridded pixel  $I_{b'}^{oblique}(k', j')$ 

#### // Regrid per pixel ortho-rectified geolocation information

// Note: this step gives strictly the same result for any channel acquired with the same sub-band ("A stripe", "B stripe" or TDI). Hence this step must be skipped if it has already been performed for a channel acquired with the same sub-band than channel b'

Read the *latitude* and *longitude* fields associated to pixel q (or equivalently S,  $p_a$ ,  $k_{det}$ ) in the *Full resolution geodetic coordinates ADS* corresponding to the oblique view sub-band of channel b' ("A stripe", "B stripe" or TDI)

Store it in an appropriate structure for oblique view ortho-geolocation data *//* **Regrid L1b clouds and confidence flags:** 

// Note: this step gives strictly the same result for any channel acquired with the same sub-band ("A stripe", "B stripe" or TDI). Hence this step must be skipped if it has already been performed for a channel acquired with the same sub-band than channel b'

In the SLSTR L1b *Global flags* ADS for oblique view corresponding to the subband of channel b' ("A stripe", "B stripe" or TDI) read:

- the *cloud* flags of spatial pixel q
- the confidence flags of spatial pixel q
- the *pointing* flags of spatial pixel q

Store them in an appropriate array (for A, B or TDI sub-band) at location (k',j') (associated to  $I_{b'}^{oblique}(k',j')$ )

#### // Obtaining L1c SLSTR oblique view 500m time-stamps from L1b ones:

// Note: this step gives strictly the same result for any channel acquired with the same sub-band ("A stripe", "B stripe" or TDI). Hence this step must be skipped if it has already been performed for a channel acquired with the same sub-band than channel b'

#### If $p_a = p_a\_ref\_SL\_ALT\_05km\_TS$ then

// Note:  $p_a_ref_SL_ALT_05km_TS$  is the absolute pixel number of a reference SLSTR oblique view 500m pixel (approximately pointing to the satellite trace).

In the SLSTR L1b *Time* ADS for oblique view corresponding to the sub-band of channel b' ("A stripe", "B stripe" or TDI) read:

- the scan\_time element corresponding to scan number S
- the *pixel\_time* element corresponding to pixel p<sub>a</sub>'

Add the two values to obtain the time-stamp of line k' and assign it to the k'<sup>th</sup> element of an array containing the SLSTR oblique view time stamps of the corresponding sub-band (A, B or TDI).

End If

End For // (loop on q) End For // (loop on b')

3.1.4.3.2.3.2 Pre-Processing for 1km channels:



**Issue:** 11 Page: 40/107

Create the L1c image structure including the 5 SLSTR oblique view 1km channels  $I_{b'}^{oblique}(k', j')$ 

for b'=7 to 11, k'=0 to N\_SCAN\_SLST\_ALT\_1km\_CUT -1, j'=0 to N\_PIX\_SCAN\_ALT\_1km-1, and set each element to PIXEL\_UNFILLED value. N\_PIX\_SCAN\_ALT\_1km is read in the SLSTR L1b product, for instance in the *Geolocation ADS* of an oblique view 1km channel it is the *n\_pixel* dimension.

#### For each channel b'=7 to 11 do

For each 1km spatial pixel q of the SLSTR oblique view in the L1b product, including orphan pixels do

Read *exception* byte (or *exception\_orphan* byte) of current pixel in the *Visible and shortwave infrared* MDS of channel b'.

If unfilled\_pixel flag is set then skip to next pixel

Read *confidence* word (or *confidence\_orphan* word) of current pixel in the *Global flags* for nadir view 1km resolution

If cosmetic flag is set then skip to next pixel

#### // Find the corresponding location (k',j') acquisition geometry:

Read the scan number S and detector number  $k_{det}$  of the oblique view spatial pixel q from the L1b product *Scan, pixel and detector number ADS, oblique view, 1km resolution* 

Compute the corresponding instrument scan number of the pixel:  $s = 2.S + k_{det}$ If  $s > s_{max}^{a,1km}$  or  $s < s_{min}^{a,1km}$  then pass directly to the next spatial pixel in the loop

Read the (relative) pixel number p<sub>a</sub>' of the oblique view spatial pixel q from the L1b product annotations (in the *Scan, pixel and detector number* ADS for oblique view 1km resolution)

Note: the formerly computed absolute pixel number  $p_n$  could be obtained as:  $p_a = p_a$ ' + FIRST\_ALONG\_TRACK\_1km\_PIXEL\_NUMBER.

// FIRST\_ALONG\_TRACK\_1km\_PIXEL\_NUMBER is the absolute pixel number of the first 1km pixel in the oblique scans (see [RD-2]), stored in the add\_offset attribute of the pixel variable

# $\mathbf{k'} = s - s_{\min}^{a,1km}$

 $j' = p_a'$ 

#### // Regrid radiometry and exception flags:

For all b'=7 to 11, read the brightness temperature (*BT*) value associated to pixel q in the oblique view *Thermal infrared* MDS corresponding to the b' channel and set  $I_{b'}^{oblique}(k', j')$  to this value

For all b'=7 to 11, read the *exception* flag associated to pixel q in the oblique view *Thermal infrared* MDS corresponding to the b' channel and attach it to the regridded pixel  $I_{b'}^{oblique}(k', j')$ 

*II* Regrid per pixel ortho-rectified geolocation information of a 1km sub-band // Note: this step is to be performed only once for one of the 1km channel, and then skipped in the other iterations

Read the *latitude* and *longitude* fields associated to pixel q (or equivalently S,  $p_n$ ,  $k_{det}$ ) in the *Full resolution geodetic coordinates ADS* of a oblique view 1km channel Store it in an appropriate structure for oblique view ortho-geolocation data *//* **Regrid L1b clouds and confidence flags:** 

In the SLSTR L1b Global flags ADS for oblique view 1km resolution read:

• the *cloud* flags of spatial pixel q



**ISSUE:** 11 Page: 41/107

- the confidence flags of spatial pixel q
- the *pointing* flags of spatial pixel q

Store them in an appropriate array at location (k',j') (associated to  $I_{b'}^{oblique}(k',j')$ ) // Obtaining L1c SLSTR oblique view 1km time-stamps from L1b ones: If  $p_a = p_a\_ref\_SL\_ALT\_1km\_TS$  then // Note:  $p_a\_ref\_SL\_ALT\_1km\_TS$  is the absolute pixel number of a reference SLSTR oblique view 1km pixel (approximately pointing to the satellite trace).

In the SLSTR L1b Time ADS for oblique view 1km resolution read:

- the scan\_time element corresponding to scan number S
- the *pixel\_time* element corresponding to pixel p<sub>n</sub>'

Add the two values to obtain the time-stamp of line k' and assign it to the k'<sup>th</sup> element of an array containing the SLSTR oblique view 1km resolution time stamps.

End If

End For // (loop on q) End For // (loop on b')

At the end of this regridding process all the elements of the  $I_{b'}^{oblique}(k', j')$  arrays must have been filled (no PIXEL\_UNFILLED value must remains).

3.1.4.3.3 Retrieving sub-sampled and general SLSTR nadir and oblique views L1b annotations

# 3.1.4.3.3.1 SLSTR sub-sampled and general L1b Annotations

The sub-sampled and general SLSTR annotations (i.e. not on the L1b pixel resolution grid) included in the level 1b product that must be handled at Level 1c are the following ones (extracted from [RD-3]):

- On the L1b product Tie Point grids (oblique and nadir view):
  - o latitude and longitude of tie point pixels (in the 16km geodetic coordinates ADS);
  - Nadir view viewing angles and satellite distance at the tie points in the 16km Solar and satellite geometry ADS for nadir view);
  - Oblique view viewing angles and satellite distance at the tie points (in the *16km Solar and satellite geometry* ADS for oblique view);
- Additional Data Sets:
  - Thermal infrared quality ADS and Visible and shortwave infrared quality ADS.

Remark: the L1b tie-points are regularly spaced on ground (on the L1b product grid). See [RD-3] and [RD-2] for a detailed description of the grids and tie-points.

3.1.4.3.3.2 Annotations Processing

# 3.1.4.3.3.2.1 Processing of the annotations on the L1b tie-point grid

Concerned annotations are:

- latitude and longitude of tie points;
- Nadir view viewing angles and satellite distance at the L1b product tie points;



- **ISSUE:** 11 Page: 42/107
- Oblique view viewing angles at the L1b product tie points;

#### Nadir view processing

The processing consists in gathering for each L1b tie-point the geolocation (lon, lat, altitude), the corresponding viewing geometry (OZA, OAA, satellite distance) to be provided in the L1c product [RD-5].

#### **Oblique view processing**

The processing consists in gathering for each L1b tie-point the geolocation (lon, lat, altitude), the corresponding viewing geometry (OZA, OAA, satellite distance) to be provided in the L1c product [RD-5].

#### 3.1.4.3.3.2.2 Additional Data Sets

There are 10 *Thermal infrared quality* ADS (S7, S8, S9, F1 and F2 for oblique and nadir view channels), 12 *Visible and shortwave infrared quality* ADS for S1 to S6 oblique and nadir view channels (the sub-bands of the SWIR channels being selected according to the SLST\_SWIR\_SELECT parameter.

The SLSTR L1b *Thermal infrared quality* ADS and *Visible and shortwave infrared quality* ADS (described in SY-4 volume 3) are partially copied in the L1c product: only the part of the variables indexed by scan numbers between  $S_{\min}^{nad}$  and  $S_{\max}^{nad}$  (resp.  $S_{\min}^{alt}$  and  $S_{\max}^{alt}$ ) defined in paragraph 3.1.4.3.2.1 are kept.

3.1.4.4 Retrieve the Sun Zenith Angle (SZA) for all instrument pixels the nadir view SLSTR  $S_u$  channel

The per pixel SZA is necessary only for nadir view 500m channels, in order to convert the SLSTR nadir view reference channel (500m) into reflectance unit (see section 3.1.4.5.2).

The data necessary to establish any of the grids is split in three datasets in the L1b products (see [RD-3]):

• Full resolution Cartesian coordinates ADS

• 16km Solar and satellite geometry ADS

The two first datasets allow retrieving the corrected location of any instrument pixel in the L1b product x/y grid. Then the corresponding latitude/longitude location is computed by interpolation in the subsampled grid included in the *Coordinate transform* ADS.

For each (k',j') of the SLSTR nadir view "A stripe" in its acquisition geometry (k'=0 to N\_SCAN\_SLST\_NAD\_05km\_CUT -1, j'=0 to N\_PIX\_SCAN\_NAD\_05km-1) do

Compute  $k_{det} = k' \mod 4$  // the (non-integer) remainder of the division of k' by 4

Compute S = (k' +  $s_{\min}^{n,05km}$  -  $k_{det}$ ) / 4 // the nadir view Scan corresponding to k'

Read the (x,y) coordinates corresponding to scan S, pixel j' and detector  $k_{det}$  in the *Full* resolution Cartesian coordinates ADS for nadir view "A stripe"



Compute the SZA of pixel (k',j') by linear interpolation at location (x,y) in the subsampled tie points array *sat\_zenith* given in the *Solar and satellite geometry* ADS in the SLSTR L1b product (see [RD-3])

Store the value in the array noted  $\theta_{SZA,SLST05km_NAD}(k', j')$ .

End For

# 3.1.4.5 Convert the radiometric unit of OLCI and SLSTR nadir view reference channels

The two following sections describes how the OLCI and SLSTR nadir view reference channels are converted from TOA radiance unit to TOA normalized radiance or TOA reflectance, depending on the value of the processing parameter UNIT\_CONV\_PARAM = 0 or 1. When UNIT\_CONV\_PARAM = 1, the reference channels are converted to TOA reflectance unit. When UNIT\_CONV\_PARAM = 0, the reference channels are converted to normalized TOA radiance unit.

3.1.4.5.1 Convert the OLCI reference channel  $O_q$ 

The algorithm is as follows:

Read the *In-band solar irradiance, seasonally corrected* for OLCI O<sub>q</sub> channel in the General Information Data file of OLCI L1b product. Due to central wavelength non-uniformity between the detectors of the channel, the data are represented in an array noted  $E_0[\lambda_q](p)$ , with p is the detector index of channel O<sub>q</sub>, p = 1 to 5 x N\_DET\_CAM.

Make a copy of the O<sub>q</sub> channel  $I_q^m(k, j)$  (5 images in radiance unit) obtained in section 3.1.4.2.2, renamed  $L_TOA_q^m(k, j)$ .

Convert  $L_TOA_q^m(k, j)$  into normalized TOA radiance or TOA reflectance unit:

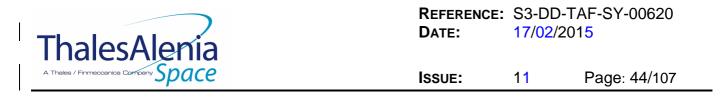
$$R_{-}TOA_{q}^{m}(k,j) = \frac{\pi . L_{-}TOA_{q}^{m}(k,j)}{E_{0}^{'^{m}}(j) . [\alpha \cos(\theta_{SZA,OLCI}^{m}(k,j)) + (1-\alpha)]}$$
eq. 3.3

where  $E_0^{m}(j) = E_0^{[\lambda_q]}((m-1)*N\_DET\_CAM+j)$  and  $\theta_{SZA,OLCI}^{m}(k,j)$  has been computed in section 3.1.4.2.2 and  $\alpha = UNIT\_CONV\_PARAM$ .

Do not convert pixel value if its L1b Invalid Pixel flag is set (or if its radiance value is set to an exceptional value by the level 1b processing).

3.1.4.5.2 Convert the SLSTR nadir view reference channel  $S_u$ 

In the SLSTR product, read the *Solar Irradiance* data in the *Visible and shortwave infrared quality* ADS for SLSTR nadir view S<sub>u</sub> channel. Noted  $E_0[\lambda_u](k_{det})$  (k<sub>det</sub> = 1 to 4). This parameter represents the "Solar Irradiance at Top-of-Atmosphere" in Wm-2nm-1, corrected for the Sun-Earth distance.



Make a copy of the nadir view  $S_u$  channel  $I_u^{nadir}(k', j')$  (in radiance unit) obtained in section 3.1.4.3.2.2.1, renamed  $L_TOA_u(k', j')$ .

Convert  $L_TOA_u(k', j')$  into normalized TOA radiance or TOA reflectance unit:

$$R_{-}TOA_{u}(k',j') = \frac{\pi . L_{-}TOA_{u}(k',j')}{E_{0}[\lambda_{u}](k_{det}(k')).[\alpha \cos(\theta_{SZA,SLST\,05km_{-}NAD}(k',j')) + (1-\alpha)]}$$
eq. 3.4

where  $\theta_{SZA,SLST 05km_NAD}(k', j')$  has been computed in section 3.1.4.4 and  $k_{det}(k') = k' \mod 4$  and  $\alpha$  = UNIT CONV PARAM.

Do not convert a pixel value if one of the flags in its L1b Exception flag is set (or if its radiance value is set to an exceptional value by the level 1b processing).

#### 3.1.4.6 Tie points selection

The tie-points selection can be obtained in several ways according to the switch TP\_SELECT\_SWITCH. In all cases, the result is 5 lists of N\_TP\_L1C(m) selected tie points in each camera module image m, noted ( $k_p$ ,  $j_p$ ), p = 1 to N\_TP\_L1C(m).

If TP\_SELECT\_SWITCH = "DB" then the tie-points are selected among those in the tie-point database as described in section 3.1.4.6.1

Else If TP\_SELECT\_SWITCH = "REGULAR\_STEP" then the tie-points are regularly sampled in the 5 OLCI camera module images as described in section 3.1.4.6.2

Else If TP\_SELECT\_SWITCH = "REGULAR\_N" then the tie-points are regularly sampled in the 5 OLCI camera module images described in section 3.1.4.6.3

Else If TP\_SELECT\_SWITCH = "EXT" then do nothing. "EXT" is set when the list of tie-points is to be read in a file (see paragraph 3.2.1.2).

#### 3.1.4.6.1 Selection of tie-points from the tie-points database:

The selection of those tie-points that lie in the area covered on ground by each of the 5 OLCI images needs that this area be determined first, using the ortho-rectified geolocation information in the L1b product. Then the ground coordinates of all the tie points in the database are compared with the domain boundaries.

For improved efficiency, the image/grid is processed by segment.

The following algorithm must be applied independently for each OLCI camera module m = 1 to 5.

The ortho-rectified geolocation grid of the considered camera module image is noted ( $\lambda_{kj}, \phi_{kj}$ ). If the grid crosses the 180° longitude meridian, then add 360° to all the negative  $\phi_{kj}$  value and to the longitude coordinate of all tie-points.



**ISSUE:** 11 Page: 45/107

The s<sup>th</sup> segment of this grid is defined as the sub-grid such that OLC\_SEGMENT\_SIZE \*  $s \le k < OLC\_SEGMENT\_SIZE$  \* (s+1), s = 0,1,2... In case the number of lines in the grid is not a multiple of OLC\_SEGMENT\_SIZE, the last segment is forced to finish at K = N\_LINE\_OLC. The Figure 3-1 shows elements to understand the algorithm described below.

#### For each segment s do

Locate the 4 corners of segment s on earth reading the ortho-rectified geolocation of these pixels. That is to say, the ortho-rectified geolocation of the following pixels:

(OLC\_SEGMENT\_SIZE \* s,0) ; (OLC\_SEGMENT\_SIZE \* s,N\_DET\_CAM) ; (OLC\_SEGMENT\_SIZE \* (s+1)-1,N\_DET\_CAM); (OLC\_SEGMENT\_SIZE \* (s+1)-1,0) The corresponding points on ground are respectively noted A B C D (see Figure 3-1)

The corresponding points on ground are respectively noted A, B, C, D (see Figure 3-1) with coordinates ( $\lambda_A, \phi_A$ ), ( $\lambda_B, \phi_B$ ), ( $\lambda_C, \phi_C$ ), ( $\lambda_D, \phi_D$ )

Compute  $\lambda_{min} = min(\lambda_A, \lambda_B, \lambda_C, \lambda_D)$ ,  $\lambda_{max} = max(\lambda_A, \lambda_B, \lambda_C, \lambda_D)$ ,  $\phi_{max} = max(\phi_A, \phi_B, \phi_C, \phi_D)$ ,  $\phi_{min} = min(\phi_A, \phi_B, \phi_C, \phi_D)$ 

Find those tie-points in the database such that:  $\lambda_{min} < \lambda_p < \lambda_{max}$  and  $\phi_{min} < \phi_p < \phi_{max}$ . The selected tie points are now indexed with g.

For each selected tie point  $(\lambda_g, \varphi_g)$ ,

Find the nearest neighbor among the locations of all the pixels in the current segment. The result is noted  $((\lambda_{kj}(g), \varphi_{kj}(g)), \text{ corresponding to pixel }(k,j))$ .

If  $j < W\_ACT\_TP\_MARGIN(m)$  or  $j > N\_DET\_CAM -1 - E\_ACT\_TP\_MARGIN(m)$  then reject the tie-point

If the current segment s is the  $1^{st}$  segment (s = 0) then

If k < ALT\_TP\_MARGIN then reject the tie-point

End if

If the current segment s is the last segment then

If k > N\_LINE\_OLC - 1 - ALT\_TP\_MARGIN then reject the tie-point

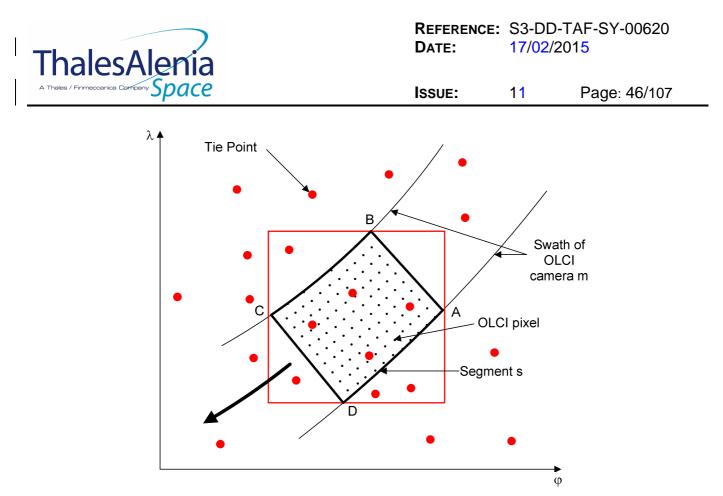
End if

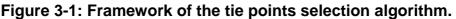
End for (loop on tie points)

Concatenate the list of tie points – and their respective nearest OLCI pixel (k,j) found previously –, for segment s obtained at each iteration into a global list for OLCI camera module m

End for (loop on segments)

Eliminate possible duplicate tie-points in the global list for camera module m image





Note: in operational condition the selection of the tie points could be done taking advantage of S3 orbital cycling, using the Orbital Revolution Number included in the NAVATTs – and that should be included in L1b products. The selection would rely on the fact that the OLCI image covers about the same area on ground at each orbit with the same Orbital Revolution Number. The tie-points database should be composed of 385 files, indexed with the Orbital Revolution Number [1 – 385]. Each file would contain a list of tie points (and annotations) that lie in the area nominally covered on ground by the OLCI image during the corresponding orbit. Thus, the processing could be:

Read the Orbital Revolution Number included in the OLCI L1b product Select the corresponding file in the tie-points database

# 3.1.4.6.2 Selection of tie-points on a regular and centered grid within the part of the $O_q^m$ image defined by margins (m = 1 to 5)

The along and across-track steps are given in input.

Figure 3-2 illustrates the result of the selection process.

Define:

- S\_ALT = N\_LINE\_OLC 2\*ALT\_TP\_MARGIN the number of rows between the margins of any camera module image
- S\_ACT(m) = N\_DET\_CAM (W\_ACT\_TP\_MARGIN(m) + E\_ACT\_TP\_MARGIN(m)) the number of columns between the margins of camera module m image

For each m = 1 to 5 do

R\_ALT = S\_ALT mod ALT\_TP\_STEP // the remainder of division of S\_ALT



**Issue:** 11 Page: 47/107

```
// by ALT TP STEP
      Q_ALT = (S_ALT - R_ALT) / ALT_TP_STEP // the quotient of division of
                                                 // S_ALT by ALT_TP_STEP
      If R ALT \geq 1 then
            L_ALT = ALT_TP_STEP * Q_ALT + 1
            N_ALT = Q_ALT + 1
                                     // number of TP in the ALT direction
      End If
      If R ALT = 0 then
            L_ALT = ALT_TP_STEP * (Q_ALT-1) + 1
            N_ALT = Q_ALT
                                     // number of TP in the ALT direction
      End If
      k_0 = FLOOR[(S_ALT - L_ALT)/2]
      R ACT = S ACT(m) mod ACT TP STEP
                                                 // the remainder of division of S ACT(m)
                                                 // by ACT_TP_STEP
      Q_ACT = (S_ACT(m) - R_ACT) / ACT_TP_STEP
                                                       // the quotient of division of
                                                        // S ACT(m) by ACT TP STEP
      If R ACT \geq 1 then
            L ACT = ACT TP STEP * Q ACT + 1
            N ACT = Q ACT + 1
                                     // number of TP in the ACT direction
      End If
      If R ACT = 0 then
            L_ACT = ACT_TP_STEP * (Q_ACT-1) + 1
                                    // number of TP in the ACT direction
            N ACT = Q ACT
      End If
      j_0 = FLOOR[(S_ACT(m) - L_ACT)/2]
      The tie-points coordinates (k_{p1,p2}, j_{p1,p2}) in camera module m image are given by:
      k_{p1,p2} = ALT_TP_MARGIN + k_0 + p_1 * ALT_TP_STEP
      j_{p1,p2} = W_ACT_TP_MARGIN(m) + j_0 + p_2 * ACT_TP_STEP
      with p_1 = 0 to N_ALT-1 and p_2 = 0 to N_ACT-1
End For
```

Tie-points over water are rejected (see section 3.1.4.6.4).

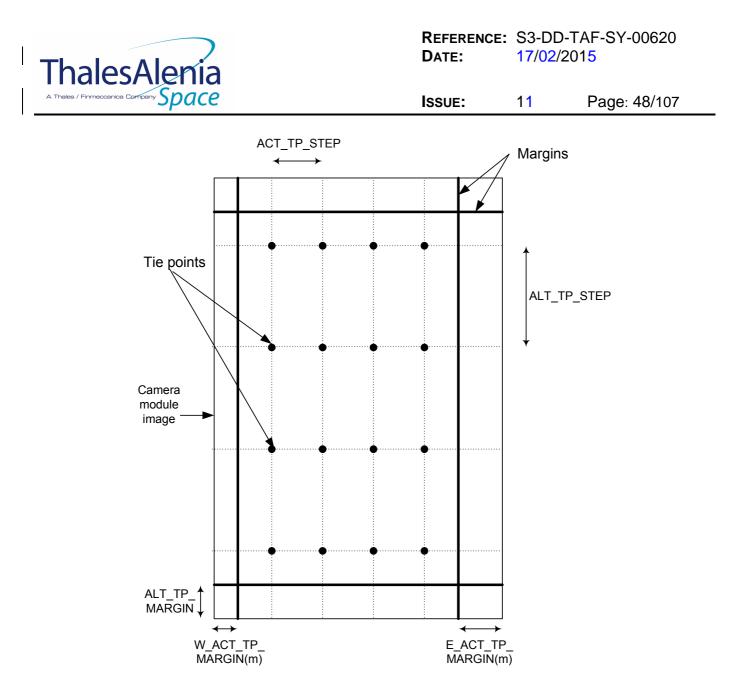


Figure 3-2: Definition of a regular grid of tie-points in a given OLCI camera module image.

3.1.4.6.3 Selection of tie-points on a regular grid within the image part of each  $O_q^m$  image defined by margins (m = 1 to 5)

The number of along and across-track tie-points are given in input.

Define:

- S\_ALT = N\_LINE\_OLC 2\*ALT\_TP\_MARGIN the number of rows between the margins of any camera module image
- S\_ACT(m) = N\_DET\_CAM (W\_ACT\_TP\_MARGIN(m) + E\_ACT\_TP\_MARGIN(m)) the number of columns between the margins of camera module m image

For each m = 1 to 5 do  $ALT\_STEP = (S\_ALT-1) / (ALT\_TP\_NUM - 1)$  $ACT\_STEP = (S\_ACT(m)-1) / (ACT\_TP\_NUM - 1)$ 



**ISSUE:** 11 Page: 49/107

The tie-points coordinates  $(k_{p1,p2}, j_{p1,p2})$  in camera module m image are given by:

 $k_{p1,p2} = ALT_TP_MARGIN + round(p_1 * ALT_STEP)$ 

 $j_{p1,p2} = W_ACT_TP_MARGIN(m) + round(p_2^* ACT_STEP)$ 

with  $p_1 = 0$  to ALT\_TP\_NUM-1 and  $p_2 = 0$  to ACT\_TP\_NUM-1

End for

Note that if ACT\_STEP or ALT\_STEP are not integers the sampling in terms of pixels can vary of  $\pm$  1 pixel along lines and rows.

Tie-points over water are rejected (see section 3.1.4.6.4).

#### 3.1.4.6.4 Rejection of (regularly) selected tie-points over water

If the corresponding flag in TP\_REJECTION\_TESTS\_SWITCHES is set, the following algorithm is to be applied to the tie-points regularly selected as described in sections 3.1.4.6.2 and 3.1.4.6.3.

For each camera module image m = 1 to 5

For each selected tie-point  $(k_p, j_p)$ , p = 1 to N\_TP\_L1C(m)

If the window of radius CW\_K\_RADIUS centered on  $(k_p, j_p)$  contains a proportion of pixels flagged "water" (in-land or ocean) greater than a threshold T\_WATER\_PIX\_TP then delete the tie-point from the list of tie points.

// The flag(s) to be checked have been retrieved in section 3.1.4.2.2 (OLCI quality flags)

End For

End For

Note: For simplicity in this document, the number of non-rejected tie-points in the list in still noted N\_TP\_L1C(m).

# 3.2 Inter-Instrument Spatial Misregistration Estimation: OLCI / SLSTR Nadir view

This module of the Level 1c processing estimates the misregistration (local shift) between one reference OLCI channel  $O_q$  and one reference SLSTR channel  $S_u$ , at each pixel location of the OLCI channel. The estimation is performed the following way:

For each tie-point selected during the pre-processing:

- 1. Extract imagettes around the tie point: one in the selected OLCI channel (the "context" imagette); one bigger in the selected SLSTR channel (the "search" imagette). In the meanwhile, the SLSTR imagette is projected to the OLCI geometry
- 2. Estimate the local shift between the two channels, at tie-point. This is achieved by computing a correlation surface between the Search imagette and the Context imagette that is shifted around the tie point according to shift vectors and finding its sub-pixel maximum.

End For

3. Estimate the parameters of a piece-wise deformation model that gives the misregistration of the reference SLSTR channel at each pixel of the OLCI selected channel, based on the misregistration measured at tie points.



4. Use the deformation model to compute the misregistration between the two channels at each OLCI pixel.

The whole Inter-Instrument Spatial Misregistration Estimation module is called 5 times, once for each camera module image of OLCI. In the sequel, the current processed OLCI channel is noted  $O_q^m$ , with m = 1 to 5.

Note: The algorithm described in this section is general enough to handle a dense correlation instead of correlation at tie points with minor modifications. In this case the tie-point database would no longer be required, as well as the deformation model (replaced by a simple interpolation in a dense grid).

#### 3.2.1 Algorithm Inputs

3.2.1.1 One SLSTR nadir band  $S_{u}$  and one OLCI FR band  $O_{q}$  selected as reference for comparison

The OLCI (resp. SLSTR) channel to be compared is extracted from the pre-processing output images (see section 3.1.3) with its annotations (mainly ortho-rectified geolocation and cloud flags).

The two reference bands to be compared are noted SLSTR  $S_u$  and OLCI  $O_q^m$  (m is the current camera module number).

#### 3.2.1.2 A list of N\_TP\_L1C(m) tie points coordinates per OLCI camera module m

The locations  $(k_p, j_p)$  of the tie-points in the  $O_q^m$  images come

- Either from the output of the Pre-processing module (if TP\_SELECT\_SWITCH = "REGULAR\_STEP" or "REGULAR\_N").
- Either from a data file with the same format as the corresponding intermediate data file output of by the pre-processing module (if TP\_SELECT\_SWITCH = "EXT"). The location of the file is given in TP\_LIST\_DATAFILE\_NAME.

#### 3.2.1.3 Processing parameters

- Interpolation parameters: for bicubic, Shannon truncated apodized and linear interpolation:
  - SW\_Interp\_Shannon\_Param
  - SW\_Interp\_Bicubic\_Param
- Geolocation and inverse geolocation functions parameters
- TP\_LIST\_DATAFILE\_NAME
- CW\_SIZE\_SWITCH
- CW\_K\_RADIUS,
- T\_SIZE\_CW
- T\_CLOUD\_PIX\_CW
- T\_LOW\_QUALITY\_FLAGS\_CW
- T\_INVALID\_PIX\_CW
- T\_GRAD\_K\_CW, T\_GRAD\_J\_CW



**Issue:** 11 Page: 51/107

- T\_GRAD\_K\_RATIO\_CW, T\_GRAD\_J\_RATIO\_CW
- T\_GRAD\_K\_SW, T\_GRAD\_J\_SW
- T\_GRAD\_K\_RATIO\_SW, T\_GRAD\_J\_RATIO\_SW
- DELTA\_SHIFT
- T\_CLOUD\_PIX\_SW
- T\_QI\_FLAGS\_SW
- T\_EXCEPTION\_FLAGS\_SW
- SW\_INTERP\_METHOD
- T\_MAX\_CORREL
- T\_CORREL\_SHAPE
- T\_MAXMEAN\_DIFF\_COR
- T\_MAXMAX\_DIFF\_COR
- DICHO\_SEARCH\_INTERP\_METHOD
- DICHO\_SEARCH\_INTERP\_SZE
- N\_ITER\_DICHO
- T\_DICHO\_CONV
- N\_TILES\_ROW
- N\_TILES\_COL
- R\_OVL\_ROW
- R\_OVL\_COL
- T\_N\_TP\_TILE
- LOC\_DEF\_MDL\_SWITCH
- LAMBDA\_TPS\_COL
- LAMBDA\_TPS\_ROW
- <u>A\_ATP\_ROW</u>
- <u>A\_ATP\_COL</u>
- MAX\_DELTA\_EST
- N\_FRAME\_MIN
- N\_TP\_MIN
- Frow
- Fcol
- TP\_REJECTION\_TESTS\_SWITCHES: Switches for each tie-points rejection

# 3.2.2 Processing Objective

The following steps (3.2.2.1, 3.2.2.2, 3.2.2.3) are processed for each one of the N\_TP\_L1C(m) input tie points, indexed with the variable p = 1 to N\_TP\_L1C(m).

# 3.2.2.1 Extraction of a couple of geometrically normalized OLCI $O_q$ / SLSTR $S_u$ imagettes around the tie point p

This processing extracts a couple of imagettes around the current tie-point in the OLCI  $O_q^m$  channel and in SLSTR S<sub>u</sub> channel. The imagette in OLCI channel is called the Context imagette. The imagette in SLSTR channel is called the Search imagette. During the processing the Context imagette is filtered (for theoretical reasons justified in [DR04]) while the Search imagette is projected to the OLCI Oqm acquisition geometry. These operations are called



"Geometric normalization". They aim at bringing the imagettes to be matched into a common reference space where the deformation is as close to a global translation as possible. It reduces distortions between the two images, in particular the strong distortion in SLSTR image due to the acquisition principle. This is done by computing direct and inverse geolocations from the L1b ortho-rectified geolocation information attached to the SLSTR and OLCI channels. After that processing the two extracted imagettes lie in a common (or at least very close)

After that processing the two extracted imagettes lie in a common (or at least very close) geometry and can be compared in the next step.

Several tests are performed on the tie-point location and context and search windows to reject the tie point if necessary. In this case tie point is deleted from the list of input tie points (nevertheless the initial list of tie points shall be stored) and step 3.2.2.1 is performed with a new tie point (extraction of a new couple of imagettes).

#### 3.2.2.2 Radiometric Normalization (not needed in baseline)

The radiometric similarity of the two selected channels is assumed to be sufficient for not being an issue concerning the L1c processing performance.

Nevertheless this section should be kept until this assumption is experimentally verified.

3.2.2.3 Sub-Pixel Local Shift Estimation at Tie Point by Matching Context and Search imagettes

In the previous stage, the geometric correspondence between the SLSTR  $S_u$  and OLCI  $O_q^m$  windows is established by means of the ortho-rectified geolocation functions that are inevitably contaminated with errors. Hence residual correspondence errors ("local shifts" in the common geometry) have to be estimated to achieve a more accurate co-registration between the two channels. This is the goal of the processing described here, which is the core of the whole Level 1c processing.

The precise estimation of the residual local shift between the Context and Search window is realized using a matching technique: the algorithm uses a sub-pixel maximization of the normalized cross-correlation coefficient as matching technique, which is known to give very good results when applied to nearly identical images.

Further tie point rejection tests are performed in this stage, based on the shape of the correlation surface.

The extracted windows must contain *salient* features in the channels to be correlated, in the sense that they have characteristics proper to give good matching results.

The tie points in the data base have been selected because they actually locate salient features on ground (see section 3.1.1.2).

#### 3.2.2.4 Estimation of Deformation Model Parameters

The local shifts estimated during the previous stage are non-densely spread out on the common grid. Then a deformation model must be fit to these data in order to be able to compute local shifts at any location on the common grid. The chosen deformation model is a piecewise linear model. It first decomposes the image into triangles by a Delaunay triangulation of the tie points augmented with a list of artificial tie-points to cover the entire image. These artificial tie-points and associated misregistration values are computed from a coarse Thin-Plate Spline (TPS)

ThalesAlenia	-	S3-DD-T 17/02/20	AF-SY-00620 1 <mark>5</mark>
A Thales / Finmeccanica Company Space	ISSUE:	11	Page: 53/107

based deformation model. This coarse TPS model is estimated from virtual tie-points obtained by "averaging" location and values of the tie points in big image areas (tiles). Secondly, within each triangle, the deformation is modeled as linear function estimated from the 3 vertex tiepoints. Finally, the model is used to compute the misregistration between the OLCI  $O_q$  channel and the SLSTR S<sub>u</sub> channel at each pixel of the  $O_q$  channel.

Hence, the deformation model is image-dependent since a new model is estimated for each new L1b product processed at level 1c.

A trade-off analysis and justifications of the deformation model can be found in [RD-6].

# 3.2.3 Algorithm Outputs

The main output of this processing module is the correspondence model between the pixels (k,j) of the OLCI  $O_q^m$  channel and the sub-pixels (k',j') of the SLSTR  $S_u$  channel (see section 3.2.4.3). It will be included in the L1c product.

The following data shall be output as intermediate verification/analysis data by the Processor:

- The list of tie points selected by pre-processing (section 3.1.3) with a flag attached to each tie point indicating by which rejection test it has been rejected during the next stage (inter-instrument Spatial misregistration estimation, section 3.2), or 0 if not rejected.
- The Context and Search imagettes at each step p, with the (kp,jp) tie point location in the OLCI Oqm channel. (Note: the O-GPP shall allow processing the 5 camera modules and the N\_TP\_L1C(m) tie points step by step at level 1c.). The (kp,jp) locations should be appended to the tie point list above.
- The correlation surface at each step p
- The maximum value of the correlation surface and the shape criterion L\*Vp used for tie point rejection (section 3.2.4.2.2)
- The  $\hat{\delta}_p^n = (\hat{\delta}_p^{row,n}, \hat{\delta}_p^{col,n})$ , for all tie-point p not rejected in a list, at each step n. And the maximum number of iterations reached (section 3.2.4.2.3)
- The estimated local shifts  $\hat{\delta}_p = (\hat{\delta}_p^{row}, \hat{\delta}_p^{col})$  at tie points p (section 3.2.4.2.3)
- The Thin-plate Spline parameters A, B, E, F
- The number of tie-points per tile with tiles centre and size (Step 2 in section 3.2.4.3.1.1)
- The list and number of triangles computed by the Delaunay triangulation
- For each triangle (from Delaunay triangulation, section 3.2.4.3.2.1), the estimated  $\alpha_u,\ \beta_u$  model parameters
- 5 masks indicating for each pixel whether the estimated (out of range) shift has been forced to (0,0) (see section 3.2.4.3.3)

# 3.2.4 Mathematical Description

3.2.4.1 Extraction of a geometrically normalized OLCI  $O_q$  / SLSTR  $S_u$  imagettes couple around the tie point p

In the sequel, for convenience purpose, the OLCI  $O_q^m$  band and the SLSTR  $S_u$  band are denominated image 1 and image 2 respectively and noted  $I_1$  and  $I_2$ . The Spatial Sampling Distances (SSD) at nadir are noted SSD<sub>1</sub> and SSD<sub>2</sub>.



The *direct* ortho-rectified geolocation functions (image point to terrain point mapping) are noted:

$(\lambda, \varphi) = \text{Loc}_1^{m}(k, j)$	For image $I_1$ (OLCI $O_q^m$ )
$(\lambda', \phi') = \text{Loc}_2(k', j')$	For image $I_2$ (SLSTR $S_u$ )

The *inverse* ortho-rectified geolocation functions (terrain point to image point mapping) are noted:

$(k,j) = (Loc_1^{m})^{-1}(\lambda,\varphi)$	For image I <sub>1</sub>
$(k', j') = Loc_2^{-1} (\lambda', \phi')$	For image I <sub>2</sub>

The direct and inverse *ortho-rectified* geolocation functions take a DEM into account to establish these (terrain/image) correspondences:  $h = DEM(\lambda, \varphi)$ .

The algorithms for computing practically the direct and inverse ortho-rectified geolocation functions from direct ortho-rectified geolocation grids are described in Annex A.

The imagettes extraction process is shown in Figure 3-3.

All the way through the process of Context and Search imagettes extraction, quality tests are performed on the imagettes possibly resulting in the rejection of the imagette. These quality tests are noted CW\_QT\_## and SW\_QT\_## for the Context and Search imagettes respectively (## will be replaced by a number). A test is performed only if the corresponding switch is set in the TP\_REJECTION\_TESTS\_SWITCHES user parameter. The tests are performed sequentially and if any of the activated tests fails:

- the next tests are not performed on the current imagette
- the current tie point is rejected
- and the next one is processed



**ISSUE:** 11 Page: 55/107

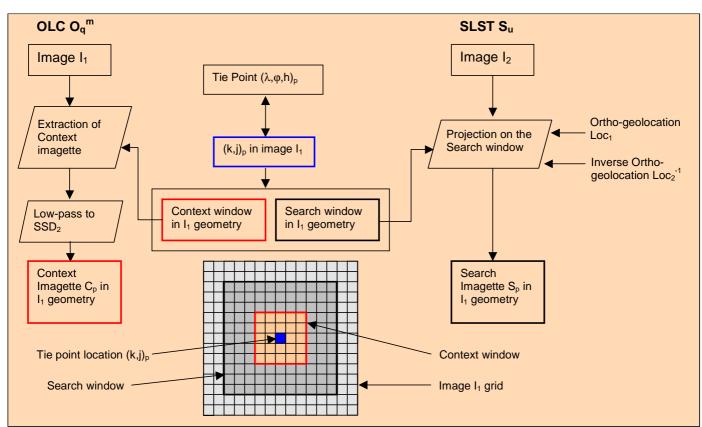


Figure 3-3: Extraction of geometrically normalized context and search imagettes around tie point p

# 3.2.4.1.1 Extraction of the context imagette

The tie points to be used by the correlation algorithm have been selected by an algorithm described in paragraph 3.1.2.4. Their location in the  $O_q^m$  channel is noted ( $k_p$ ,  $j_p$ ).

The size (in OLCI pixels) of the Context Window is now defined:

```
If TP_SELECT_SWITCH = "DB" Then
```

```
If CW_SIZE_SWITCH == "FIXED" then // use user-defined fixed Context window size d_p = CW_K_RADIUS
```

End if

```
If CW_SIZE_SWITCH == "AUTO" then // Compute automatically the size of the Context // Window
```

The radius of the Context window given in the tie points database must be converted from ground distance unit to a number  $d_p$  of OLCI pixels  $O_q^m$ :  $d_p \approx round(radius(C_p)/PS(m, k_p, j_p))+1$ .

The local OLC SSD PS(m,  $k_p$ ,  $j_p$ ) is obtained as follows:

- Read the ortho-rectified geo-location of the two adjacent pixels (k<sub>p</sub>, j<sub>p</sub>) and (k<sub>p</sub>+1, j<sub>p</sub>+1). This gives (λ<sub>p</sub>,φ<sub>p</sub>) and (λ<sub>p</sub>',φ<sub>p</sub>')
- Compute the geodetic distance on ground (at altitude 0 on the WGS84 Ellipsoid) between  $(\lambda_p, \phi_p)$  and  $(\lambda_p', \phi_p')$ . The geodetic distance can be computed using Vincenty's formula (see links to reference in section A.2.3).



**ISSUE:** 11 Page: 56/107

End if Else // (TP\_SELECT\_SWITCH  $\neq$  "DB") CW\_SIZE\_SWITCH == "FIXED"  $d_p = CW_K_RADIUS$ End If

End If

Quality test 1:

**CW\_QT\_1**: If the size (in pixel) of the Context window is too small (threshold T\_SIZE\_CW), then reject the tie point and perform step 3.2.4.1.1 with a new tie point. The rule is: if  $d_p < T_SIZE_CW$ , reject the tie point p. T\_SIZE\_CW is a fixed threshold.

The context imagette C<sub>p</sub> centered on the tie point p has an odd diameter D<sub>p</sub> = 2d<sub>p</sub> + 1 and is defined as W(C<sub>p</sub>) =  $\{(k_p - d_p + k, j_p - \Delta + j), k, j = 0, 1, ..., D_p - 1\}$ .

Before the imagette is extracted, a low-pass filter is applied to the image around the tie point  $(k_p, j_p)$  so that the spatial frequency range in the extracted imagette is limited to  $(SSD_2)^{-1}$ , similar to the one of the SLSTR S<sub>u</sub> channel. The low-pass filter to be used is a product between a cardinal sine and an apodization function in the spatial domain. Let  $r = SSD_2/SSD_1$ . The radius of the filter, in image I<sub>1</sub> pixels, is ws = round(8r). The 1D expression of the filter h<sub>1D</sub> is:

for any integer g in [-ws,ws], 
$$h_{1D}(g) = \frac{1}{r} \operatorname{sinc}\left(\frac{g}{r}\right) W(g)$$

with  $\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$  (sinc(0)=1)

and W is a Blackman Harris 4 terms window (-74dB):

$$W(g) = \alpha_{1} + \alpha_{2} \cos(2\pi(g + ws)/(2ws)) + \alpha_{3} \cos(4\pi(g + ws)/(2ws)) + \alpha_{4} \cos(6\pi(g + ws)/(2ws))$$
where
$$\begin{cases} \alpha_{1} = 0.40217 \\ \alpha_{2} = -0.49703 \\ \alpha_{3} = 0.09392 \\ \alpha_{4} = -0.00183 \end{cases}$$

The 2D low-pass filter  $h_{2D}$  is separable. Thus, the convolution of an image I(k,j) with  $h_{2D}$  is:

$$M(k, j) = \sum_{m=-ws}^{ws} h_{1D}(m) \left[ \sum_{n=-ws}^{ws} h_{1D}(n) I(k-m, j-n) \right]$$

being carefull that k-m and j-n are still in the image. Note: the filter coefficients  $h_{1D}(n)$ , n=-ws...ws should be stored in a table.

The filter must be applied to all pixels in the Context window, avoiding edge effect. Hence, the radius of the image area considered for filtering should be greater than  $d_p$  + ws around. Then, the central part is extracted resulting in the context imagette  $C_p$  centered on the tie point p, defined as  $C_p(k, j) = I_1(k_p - d_p + k, j_p - d_p + j)$ , for  $k, j = 0, 1, ..., D_p - 1$ .

Quality test 2:



**CW\_QT\_2**: If the Context window augmented with a ws pixels strip around the window contains a proportion of cloudy pixels above a certain threshold T\_CLOUD\_PIX\_CW, the misregistration will not be measured at this tie point. Then the tie point is deleted from the list of input tie points and step 3.2.4.1.1 is performed with a new tie point.

On the same area, tests based on the L1b quality flags attached to each pixel (retrieved in section 3.1.4.2) must be performed to reject the tie point if the radiometric quality of the Context imagette is poor.

**CW\_QT\_3**: If the Context window augmented with a ws pixels strip around the window contains a proportion of pixels with *Invalid Pixel* flag sets above a certain threshold T\_INVALID\_PIX\_CW then delete the tie point from the tie point list and perform step 3.2.4.1.1 with a new tie point.

**CW\_QT\_4**: If the Context window augmented with a ws pixels strip around the window contains a proportion of pixels with *Cosmetic OR Dubious OR Saturated Pixel* flag sets above a certain threshold T\_LOW\_QUALITY\_FLAGS\_CW then delete the tie point from the tie point list and perform step 3.2.4.1.1 with a new tie point.

Quality test 5: Test on the gradient of the imagette.

CW\_QT\_5:

At each pixel of the Context imagette compute the components of the gradient along lines and columns  $F_k(k,j)$  and  $F_j(k,j)$  defined as  $F_k(k,j) = C_p(k+1,j) - C_p(k,j)$  and  $F_j(k,j) = C_p(k,j+1) - C_p(k,j)$ .

Compute:

• the number  $N_K$  of pixels (k,j) where  $abs(F_k(k,j)) \ge T\_GRAD\_K\_CW$ 

• the number N<sub>J</sub> of pixels (k,j) where  $abs(F_i(k,j)) \ge T_GRAD_J_CW$ 

If  $N_K / D_p^2 < T_GRAD_K_RATIO_CW OR N_J / D_p^2 < T_GRAD_J_RATIO_CW$  then reject the current tie point from the list of tie points and process the next tie point.

3.2.4.1.2 Extraction of the Search imagette  $S_p$ 

Let  $\{(\delta^{row}, \delta^{col}), \delta^{row}, \delta^{col} = -\Delta, -\Delta + 1, ..., \Delta - 1, \Delta\}$  be a set of shift vectors ( $\Delta$  = DELTA\_SHIFT is a fixed positive integer) in the common  $O_q^m$  grid.

The "Search window"  $W(S_p)$  centered at  $(k_p, j_p)$  in image  $I_1$  is defined by sliding the Context window over  $(k_p, j_p)$ , along the translation vectors  $(\delta_k, \delta_j)$ . It can be written:

$$W(S_p) = \{(k_p - d_p - \Delta + k, j_p - d_p - \Delta + j), k, j = 0, 1, ..., 2\Delta + D_p - 1\}$$

Its diameter is  $R_p = 2\Delta + D_{p.}$ 

The corresponding grid in image  $I_2$  is found using direct and inverse geolocation of image  $I_1$  and image  $I_2$ . Indeed, for all pixels (k,j) in the Search window  $S_p$ , we want to find coordinates (k'<sub>kj</sub>, j'<sub>kj</sub>) in image 2 such that:

$$Loc_2 (k'_{kj}, j'_{kj}) = Loc_1^m (k_p - d_p - \Delta + k, j_p - d_p - \Delta + j)$$
 eq. 3-2

Using the inverse geolocation functions this is achieved by computing

$$(k'_{kj}, j'_{kj}) = Loc_{2}^{-1} \circ Loc_{1}^{m} (k_{p} - d_{p} - \Delta + k, j_{p} - d_{p} - \Delta + j).$$
eq. 3-3



for search imagette coordinates  $k_i = 0, 1, ..., R_p$ -1. The  $\circ$  symbol is the composition function.

In the following,  $Loc_2^{-1} \circ Loc_1^m$  will be noted  $G_{12}$ .  $G_{12}$  establishes a correspondence between image 1 and image 2 coordinates. The more precise are the geolocation functions, the more precise is the correspondence. We also define  $G_{21} = (Loc_1^m)^{-1} \circ Loc_2$  and we have the property  $G_{12}^{-1} = G_{21}$ .

The algorithm is as follows:

For each pixel (k,j) in the Search imagette do // the order in which the pixels are processed

// depends on the initialization process (see
// below)

Read its ortho-rectified geolocation ( $\lambda_{kj}$ ,  $\varphi_{kj}$ ) // This is Loc<sub>1</sub><sup>m</sup>

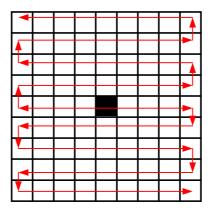
Compute the inverse geolocation  $(k'_{kj}, j'_{kj}) = Loc_2^{-1}(\lambda_{kj}, \varphi_{kj})$  in the SLSTR S<sub>u</sub> channel (see Annex A). The ortho-rectified geolocation grid of the SLSTR S<sub>u</sub> channel is passed to the function. See below for the initialization of the function.

End For

Note: If the inverse geolocation function does not converge or outputs an error the tie-point is rejected and a new one processed.

Initialization of the inverse geolocation function:

- If the processed pixel (k,j) is the first pixel of the list (for instance the central pixel of the Search imagette) then find the nearest neighbor of  $(\lambda_{kj}, \phi_{kj})$  in the ortho-rectified geolocation grid of the SLSTR S<sub>u</sub> channel and consider the corresponding (k',j') coordinates in this grid as a first guess for the inverse geolocation solution.
- For any other pixel in the Search imagette the corresponding (k',j') first guess is the result of the correspondence function previously computed for a neighboring pixel in the Search imagette. Hence, the pixels must be processed gradually, from one pixel to its neighbor(s). The Figure 3-4 shows an example of possible processing order of pixels starting from the central pixel.



# Figure 3-4: Example of pixels processing order profiting of adjacency of pixels in order to optimize the inverse geolocation processing in the Search imagette. Only the first (central) pixel needs a dedicated initialization algorithm.

Let define the "square envelope" of the Search window by its upper left and bottom right corners, respectively (k'<sub>min</sub>, j'<sub>min</sub>) and (k'<sub>max</sub>, j'<sub>max</sub>), with:



**Issue:** 11 Page: 59/107

 $k'_{\min} = Round[\min_{k,j} k'_{kj}]$  $k'_{\max} = Round[\max_{k,j} k'_{kj}]$  $j'_{\min} = Round[\min_{k,j} j'_{kj}]$ 

 $j'_{\max} = Round[\max_{k \in j} j'_{kj}]$ 

Quality test 1:

**SW\_QT\_1**: If the square envelope of the Search window augmented with a pixels strip (whose width is equal to the radius of the radiometric interpolation window used below) contains a proportion of cloudy pixels (i.e. with the *summary cloud* confidence flag set to 1) above a certain threshold T\_CLOUD\_PIX\_SW, the misregistration will not be measured at this tie point. Then the tie point is deleted from the list of input tie points and step 3.2.4.1.1 is performed with a new tie point.

On the same area, tests based on the L1b quality (exception and confidence) flags attached to each pixel (retrieved in section 3.1.4.3) must be performed to reject the tie point if the radiometric quality of the search imagette is poor.

**SW\_QT\_2**: If the square envelope of the Search Window augmented with a pixels strip (whose width is equal to the radius of the radiometric interpolation window used below) contains a proportion of pixels with QI\_FLAG sets above a certain threshold T\_QI\_FLAGS\_SW then delete the tie point from the tie point list and perform step 3.2.4.1.1 with a new tie point. QI\_FLAG is defined as *sun\_glint* flag OR *snow* flag OR *saturation* flag. *sun\_glint* and, *snow* are confidence flags corresponding to the nadir view S<sub>u</sub> channel. *saturation* flag is an exception flag corresponding to the nadir view S<sub>u</sub> channel.

**SW\_QT\_3**: If the square envelope of the Search Window augmented with a pixels strip (whose width is equal to the radius of the radiometric interpolation window used below) contains a proportion of pixels with EXCEPTION\_FLAG sets above a certain threshold T\_EXCEPTION\_FLAGS\_SW then delete the tie point from the tie point list and perform step 3.2.4.1.1 with a new tie point. EXCEPTION\_FLAG is defined as a logical OR of all the exception flags of the pixel except the *Saturation in Channel* flag.

Finally the Search imagette is computed by resampling  $I_2$  on the grid defined by eq. 3-3: ( $k'_{kj}$ ,  $j'_{kj}$ ) indexed by the search window coordinates (k,j).

At each grid point  $(k'_{kj}, j'_{kj})$  in image 2 the new pixel value is found by bicubic (see Annex B) or Shannon truncated apodized kernel (see [B99] and Annex C) interpolation on the grid defined by  $(k_p - d_p - \Delta + k, j_p - d_p - \Delta + j)$  with  $k,j = 0,1,...,2\Delta + D_p$ -1. The choice of the interpolation method is given by the SW\_INTERP\_METHOD switch. In the case of a Shannon truncated apodized interpolation (with e.g. Hann or Hamming window), the size of the kernel is 2Nsha+1 pixels. If the interpolation window is not fully contained in the image then use the biggest window still included in the image (a warning message should be logged/sent to the user).

The search imagette is in the same geometry as the context imagette. We have the relation  $S_p(k,j) = I_2(k'_{kj}, j'_{kj})$ , where  $(k'_{kj}, j'_{kj})$  is given by eq. 3-3.



**ISSUE:** 11 Page: 60/107

$$S_p(k, j) = I_2(G_{12}(k_p - d_p - \Delta + k, j_p - d_p - \Delta + j)), \text{ for } k, j = 0, 1, ..., 2\Delta + D_p - 1$$

Figure 3-5 shows the sequence of operations for computing the search imagette from the search window, in the case of a bicubic interpolation.

Quality test 4: Test on the gradient of the imagette. To be performed only if the corresponding switch is set.

# SW\_QT\_4:

At each pixel of the Search imagette compute the components of the gradient along lines and columns  $F_k(k,j)$  and  $F_j(k,j)$  defined as  $F_k(k,j) = S_p(k+1,j) - S_p(k,j)$  and  $F_j(k,j) = S_p(k,j+1) - S_p(k,j)$ .

Compute:

- the number  $N_K$  of pixels (k,j) where  $abs(F_k(k,j)) \ge T_GRAD_K_SW$
- the number N<sub>J</sub> of pixels (k,j) where  $abs(F_i(k,j)) \ge T_GRAD_J_SW$

If  $N_K / R_p^2 < T_GRAD_K_RATIO_SW OR N_J / R_p^2 < T_GRAD_J_RATIO_SW$  then reject the current tie point from the list of tie points and process the next tie point from the start (section 3.2.4.1.1).

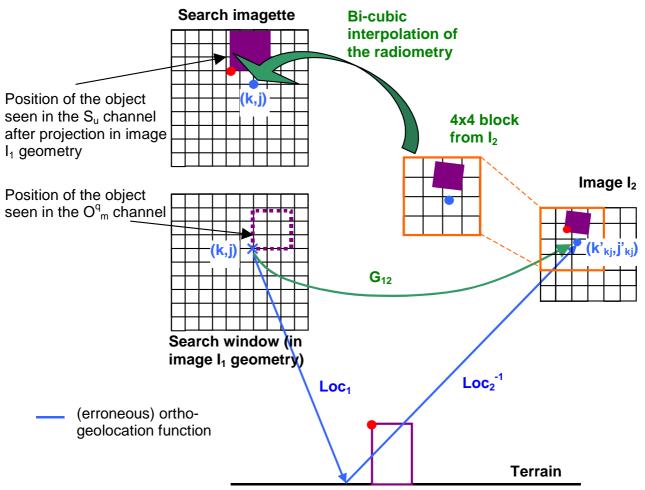


Figure 3-5: Projection of SLSTR S<sub>u</sub> Band (image 2) to OLCI O<sub>q</sub><sup>m</sup> Geometry (search imagette). In the scene a parallelepipedal object is represented in purple. It appears with



#### **Issue:** 11 Page: 61/107

#### different shapes in the two images $I_1$ and $I_2$ in their acquisition geometry. After projection of the $I_2$ image to the $I_1$ geometry, a misregistration residual appears (gap between the red and blue point in the search imagette). This residual is estimated at level 1c by a correlation-based method.

# 3.2.4.2 Local Shifts Estimation by Matching Context and Search Windows

In this stage the residual local shift between Context and Search imagettes is estimated in the common grid. The proposed matching algorithm uses correlation and can be split into two steps:

- computation of the correlation surface
- computation of the sub-pixel correlation maximum
- A tie-point rejection is also performed.

# 3.2.4.2.1 Feature Matching: Building the Correlation Surface

Figure 3-6 illustrates the relation between Search and Context windows.

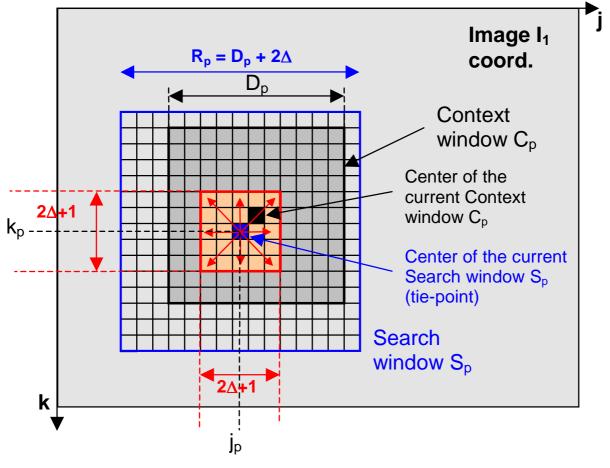


Figure 3-6: Search and Context Windows. Here  $\Delta$  = 2 pixels. Current shift between Context window and Search Window is ( $\delta^{row}$  = -1,  $\delta^{col}$  = 1) pixels. D<sub>p</sub> = 11 pixels, R<sub>p</sub> = 15 pixels.



For each shift vector  $\delta = (\delta^{\text{row}}, \delta^{\text{col}})$  a similarity measure is computed between the shifted Context imagette and the Search imagette area below the context imagette. The similarity measure used is the cross-correlation coefficient.

The cross-correlation coefficient between Context imagette  $C_p$  and Search imagette  $S_p$ , computed at  $(k_p, j_p)$  for shift  $(\delta^{row}, \delta^{col})$  is:

$$\rho_{p}(\delta^{row},\delta^{col}) = \frac{\sum_{m=0}^{D_{p}-1} \sum_{n=0}^{D_{p}-1} (C_{p}(m,n) - \mu_{c})(S_{p}(m+\Delta+\delta^{row},n+\Delta+\delta^{col}) - \mu_{s}(\delta^{row},\delta^{col}))}{\left[\sum_{m=0}^{D_{p}-1} \sum_{n=0}^{D_{p}-1} (C_{p}(m,n) - \mu_{c})^{2}\right]^{1/2} \left[\sum_{m=0}^{D_{p}-1} \sum_{n=0}^{D_{p}-1} (S_{p}(m+\Delta+\delta^{row},n+\Delta+\delta^{col}) - \mu_{s}(\delta^{row},\delta^{col}))^{2}\right]^{1/2}} \quad \text{eq. 3-4}$$

for  $\delta^{row}, \delta^{col} = -\Delta, -\Delta + 1, ..., \Delta - 1, \Delta$ 

 $\mu_c$  is the radiometric mean of the Context imagette.

 $\mu_{s}(\delta^{ow}, \delta^{col})$  is the radiometric mean of the Search imagette *in the region under the context imagette*.

In image-like coordinates the correlation surface will also be noted (abusively)  $\rho_p(u,v)$ , with  $u = \Delta + \delta^{col}$ . Then u,v can take the values  $0, 1...2\Delta$ .

Since it is computed from the imagettes in  $I_1$  geometry the cross-correlation surface is natively sampled at SSD<sub>1</sub> at nadir on the common grid.

The cross-correlation coefficient varies between -1 and 1. The closer to 1 the coefficient, the more similar the two context windows will be<sup>4</sup>. Consequently, at each tie point p, the local misregistration shift  $\delta_p = (\delta_p^{row}, \delta_p^{col})$  can be estimated by computing:

$$\hat{\delta}_p = \operatorname*{argmax}_{\delta} \rho_p(\delta)$$
 eq. 3-5

At this stage of the processing only a pixel precision can be achieved. A sub-pixel precision is obtained in the subsequent processing stage (section 3.2.4.2.3).

The diameter of this correlation surface  $\rho_p$  (matrix) is  $[2\Delta+1,2\Delta+1]$  and depends on the amplitude of the shift vectors, i.e. on the amplitude of the misregistration to be measured. So the algorithm is able to measure a shift from  $-\Delta$  to  $+\Delta$  pixels between the two imagettes.

Note about the algorithmic complexity: the spatial cross-correlation requires on the order of  $D^2\Delta^2$  operations. The numerator can be computed via the Fast Fourier Transform (FFT), which may be more efficient than direct spatial correlation.

3.2.4.2.2 Tie Points Rejection on cross-correlation results

<sup>&</sup>lt;sup>4</sup> When the coefficient is close to -1 the two imagettes are "anti-correlated", i.e. with opposite contrast, which is not expected and considered negative if Sref and Oref cover similar spectral domains.

ThalesAlenia	Reference: Date:	: S3-DD-TAF-SY-00620 17/02/2015	
A Thales / Finmeccanica Company Space	ISSUE:	11	Page: 63/107

At some Tie Points the similarity measure can be of very poor quality, which may lead to important errors on the estimation of the misregistration. In order to preserve an overall accuracy of local shifts estimation the worst Tie Points have to be rejected.

Several criteria are used to assess the quality of a matching:

- Value of the cross-correlation coefficient at maximum compared to a threshold T\_MAX\_CORREL
- Shape of the correlation surface near the maximum: if the surface is nearly flat around its maximum, then the location of the maximum is not very accurate. This criterion is based on the laplacian of the correlation surface (at SSD<sub>1</sub>) around the found maximum. It can be computed by making the convolution between the 3x3 neighbourhood of the maximum and the following kernel:

$$\mathsf{L} = \frac{1}{4} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$
 eq. 3-6

- Difference between the maximum and mean value of the cross-correlation coefficient surface
- Possible presence of a second maximum over the correlation surface

The corresponding rejection rules are as follows:

- If  $\rho_p(\hat{\delta}_p) < T_MAX_CORREL$  then reject the tie point p
- If  $L * V_p < T\_CORREL\_SHAPE$  then reject the tie point *p*. Where  $V_p$  is the 3x3 neighbourhood extracted around the maximum  $\hat{\delta}_p$  of the correlation surface (at SSD<sub>1</sub>). The symbol \* stands for convolution product (element by element product then summation of all elements). T\_CORREL\_SHAPE is a threshold.
- If  $\rho_p(\hat{\delta}_p)$  Mean( $\rho_p(\delta)$ ) < T\_MAXMEAN\_DIFF\_COR then reject the tie point p
- If  $\rho_p(\hat{\delta}_p)$  max(*lim\_p*) < T\_MAXMAX\_DIFF\_COR then reject the tie point *p*. *lim\_p* represents the correlation surface without a 3x3 neighborhood around  $\hat{\delta}_p$ .

Notes:

- Only the tests whose corresponding switches are set in the TP\_REJECTION\_TESTS\_SWITHES variable are performed.
- If <u>any</u> of the tests performed fails then the tie-point is rejected.
- In practice, tests are performed sequentially, thus when one test fails the tie point is rejected and the other tests are not performed.

The final number of tie-points in camera module image m to be used in next processing steps is noted N\_TP\_L1C\_END(m).

# 3.2.4.2.3 Sub-pixel local shifts estimation

Note: The term subpixel precision is erroneous here, since the data to be matched have different pixel sizes. The term sub "correlation surface sample" would be more exact.



#### 3.2.4.2.3.1 General principles of the method

Under the premise that the correlation matrix samples a continuous correlation surface, subpixel local shifts estimation consists in locating the maximum of this continuous correlation surface. In principle, this is achieved by interpolating (zooming) the surface around its discrete maximum. The integer zooming factor is noted  $Z_COR$ . The numerator and denominator of the cross-correlation coefficient are interpolated separately (justification appears in [DR04]).

Hence, if eq. 3-4 is re-written (with obvious notation correspondences) as:

$$\rho(\delta^{row}, \delta^{col}) = \frac{N(\delta^{row}, \delta^{col})}{\sqrt{V_c \cdot V_s(\delta^{row}, \delta^{col})}}$$
eq. 3-7

the interpolated cross-correlation coefficient is constructed by interpolating  $N(\delta^{row}, \delta^{col})$  and  $V_s(\delta^{row}, \delta^{col})$  and making the interpolated quotient

$$\rho(\delta^{row,o}, \delta^{col,o}) = \frac{N(\delta^{row,o}, \delta^{col,o})}{\sqrt{V_c \cdot V_s(\delta^{row,o}, \delta^{col,o})}}$$
eq. 3-8

with  $\delta^{row,o} = -\Delta + u/Z \_COR$ ,  $\delta^{row,o} = -\Delta + v/Z \_COR$  (the exponent *o* stands for oversampled) and  $u, v = 0, 1, ..., 2\Delta .Z \_COR$ . Note that the term V<sub>c</sub> is constant with respect to the shift and then can be omitted during the maximum search.

The interpolation method will be bi-linear or bi-cubic (depending on user parameter DICHO\_SEARCH\_INTERP\_METHOD). The accuracy of the maximum estimation depends on the resolution (linked to  $Z_{COR}$ ) of the sampling grid on which the correlation surface is interpolated.

#### 3.2.4.2.3.2 Practical implementation

Practically, the processing load is improved by dichotomic search on the interpolated crosscorrelation map, using successive zooms of factor 2 on constant size areas (4Lo+1) at each iteration. The objective is to achieve a final zooming factor equal to the desired one (Z\_COR) but localized around the maximum (i.e. without interpolating the entire correlation surface). See Figure 3-7. In this iterative scheme, at each step n only an increasingly small neighbourhood (of equivalent size  $(2L_0+1)/2^{n-1}$  pixels in the *initial* correlation surface) of the current maximum is zoomed by an increasing zooming factor  $2^n$  and the local maximum of the zoomed surface is estimated.  $L_0 = DICHO_SEARCH_INTERP_SZE$ .

The dichotomic algorithm starts with the location  $\hat{\delta}_p = (\hat{\delta}_p^{row}, \hat{\delta}_p^{col})$  of the maximum pixel of the correlation surface found with eq. 3-5.

#### • Iteration 1:

A (2L<sub>0</sub>+1) x (2L<sub>0</sub>+1) pixels neighbourhood of  $\hat{\delta}_p$  is extracted and zoomed by a factor 2. The resulting sub-pixel grid of size (4L<sub>0</sub>+1) x (4L<sub>0</sub>+1) is noted:

Neigh(1) = {
$$(\delta^{row,1}, \delta^{col,1}),$$
  
with  $\delta^{row,1} = \hat{\delta}_p^{row} - L_0 + u/2, \ \delta^{col,1} = \hat{\delta}_p^{col} - L_0 + v/2 \text{ with } u, v = 0, 1, ..., 4L_0$ } eq. 3-9



The values  $N(\delta^{row,1}, \delta^{col,1})$  and  $V_s(\delta^{row,1}, \delta^{col,1})$  are obtained by an interpolation on Neigh(1) of the *initial* correlation surface described in section 3.2.4.2.1. The interpolation method (bicubic or bilinear) is defined by the DICHO\_SEARCH\_INTERP\_METHOD parameter. Note: the implementation shall perform the zoom/interpolation on a larger window, to avoid edge effect...

Then  $\rho_p(\delta^{row,1}, \delta^{col,1})$  is computed using eq. 3-8 and the (sub-pixel) location of the maximum is found:  $\hat{\delta}_p^1 = (\hat{\delta}_p^{row,1}, \hat{\delta}_p^{col,1}) = \underset{\text{strong total}}{\operatorname{argmax}} \rho_p(\delta^{row,1}, \delta^{col,1})$ 

#### • Iteration n:

A (2L<sub>0</sub>+1) x (2L<sub>0</sub>+1) pixels neighbourhood of  $\hat{\delta}_p^n$  is extracted and zoomed by a factor 2<sup>n</sup>. The resulting sub-pixel grid of size (4L<sub>0</sub>+1) x (4L<sub>0</sub>+1) is noted:

Neigh(n) = { $(\delta^{row,n}, \delta^{col,n}),$ with  $\delta^{row,n} = \hat{\delta}_p^{row,n-1} - L_0 / 2^{n-1} + u / 2^n, \ \delta^{col,n} = \hat{\delta}_p^{col,n-1} - L_0 / 2^{n-1} + v / 2^n \text{ with } u, v = 0, 1, ..., 4L_0$ } eq. 3-10

The values  $N(\delta^{row,n}, \delta^{col,n})$  and  $V_s(\delta^{row,n}, \delta^{col,n})$  are obtained by an interpolation on Neigh(n) of the terms of the *initial* correlation surface described in section 3.2.4.2.1. The interpolation method (bicubic or bilinear) is defined by the DICHO\_SEARCH\_INTERP\_METHOD parameter. Note: the implementation shall perform the zoom/interpolation on a larger window, to avoid edge effect...

Then  $\rho_p(\delta^{row,n}, \delta^{col,n})$  is computed using eq. 3-8 and the (sub-pixel) location of the maximum is found:  $\hat{\delta}_p^n = (\hat{\delta}_p^{row,n}, \hat{\delta}_p^{col,n}) = \underset{\delta^{row,n}, \delta^{col,n}}{\operatorname{asgmax}} \rho_p(\delta^{row,n}, \delta^{col,n})$ 

The zooming factor at iteration n is  $z = 2^n$ , thus  $Z\_COR = 2^{N\_ITER\_DICHO}$ . The total number of interpolations required by the dichotomic method after n steps is 4n x 9 while it is  $4^n \times 9$  for a global interpolation method. The dichotomic method can be implemented as a recursive procedure.

#### • Stop Criteria:

The iterations stop when:

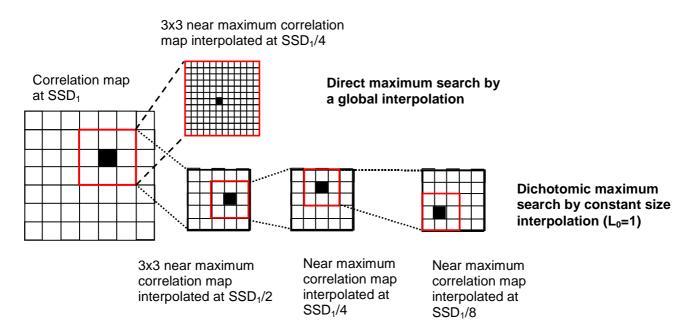
- the maximum number of iterations N\_ITER\_DICHO is reached
- **OR** when a convergence criterion is met. The criterion is based on the mean value of the gradient around maximum:
  - At iteration n, compute the convolution (element by element product then summation of all elements) between L (defined in eq. 3-6) and the 3x3 neighbourhood of the (zoomed) correlation surface  $\rho_p(\delta^{row,n}, \delta^{col,n})$  around

maximum  $\hat{\delta}_p^n$ . The result is noted  $G_p^n$ .

• Stop iteration if:  $2^n \cdot G_p^n < T_DICHO_CONV$ 



**ISSUE:** 11 Page: 66/107



# Figure 3-7: Correlation maximum localization by direct and dichotomic interpolation

Finally as an output of this stage one have a new *sub-pixel* estimation of the misregistration vector  $\hat{\delta}_p = (\hat{\delta}_p^{row}, \hat{\delta}_p^{col})$  at current tie point  $(k_p, j_p)$ .

# 3.2.4.2.4 Tie Points Rejection

The objective of this last test is to reject tie points that have been detected as aberrant. An aberrant tie point is a tie point for which its sub-pixel shift along row ( $\hat{\delta}^{row}$ ), or along column ( $\hat{\delta}^{cvol}$ ) differs significantly from the general trend of the group of tie points.

This rejection test applies independently on group of tie points that have passed all the previous rejection tests, and that are located in a block of data corresponding of a user configurable parameter indicating the minimun number of OLCI frames ("N\_FRAME\_MIN"). The initial group starts from the first frame of OLCI (output of pre-processing). For the last group, the number of frames can be higher in order to consider all the tie points (case where the number of OLCI frames is not a multiple of the minimum of OLCI frames). There is no overlaping between the groups.

This test applies if the number of tie points in the group is sufficiently high. The minimum number of tie points is a user configurable parameter ("N\_TP\_MIN").

A Tie point is considered aberrant when its shift along the row lies outside the following domain:  $[Median(\hat{\delta}_{p_{g}}^{row})-Frow^{*} \mathbf{O}(\hat{\delta}_{p_{g}}^{row}); Median(\hat{\delta}_{p_{g}}^{row})+Frow^{*} \mathbf{O}(\hat{\delta}_{p_{g}}^{row})].$ 

or when its shift along the column lies outside the following domain:



**ISSUE:** 11 Page: 67/107

 $[\text{Median}(\hat{\delta}_{P_s}^{cvol})\text{-}\text{Fcol}^* \mathbf{O}(\hat{\delta}_{P_s}^{cvol}); \text{Median}(\hat{\delta}_{P_s}^{cvol})\text{+}\text{Fcol}^* \mathbf{O}(\hat{\delta}_{P_s}^{cvol})].$ 

Where:

- Median(δ<sup>row</sup><sub>p<sub>s</sub></sub>) (repectively Median(δ<sup>cvol</sup><sub>P<sub>s</sub></sub>)) are the median value of the shift in row (repectively shift in column) of the group of tie points (g=1 to G with G = number of tie points in the group that is ≥ "N\_TP\_MIN").
- σ(δ<sup>row</sup><sub>p<sub>s</sub></sub>) (respectively σ(δ<sup>cvol</sup><sub>p<sub>s</sub></sub>)) are the standard deviation of the shift in row (repectively shift in column) of the group of tie points (g=1 to G with G= number of tie points in the group that is ≥ "N\_TP\_MIN").
   ;
- Frow (respectively Fcol) is the multiplicator factor (potitive real value) dedicated to the shift in row (respectively shift in column).

A tie point that has been detected as aberrant is rejected.

# 3.2.4.3 Deformation Model Estimation

Contrary to the previous processing, this processing is performed inside the loop on each OLCI camera module but outside the loop on the tie points. It takes as input all the tie-points used previously (i.e. not rejected) plus the local misregistration measurements estimated above.

The deformation model establishes a correspondence between image  $I_1$  ( $O_q^m$ ) and Image  $I_2$  ( $S_u$ ) for all pixels, based on the misregistration vectors estimated at tie points.

3.2.4.3.1 Estimation of a preliminary smooth Deformation Model

A first deformation model is estimated over the whole image  $I_1$  taking into account only smooth (averaged) variations of the misregistration field. The well known Thin-Plate Spline (TPS) model is chosen for this purpose.

3.2.4.3.1.1 Generation of a loose net of averaged tie-points

# Step 1:

The image  $I_1$  is first broken down into regularly spaced and identical overlapping tiles as shown in Figure 3-8. The size  $L_r \times L_c$  and the positions of the centers of the tiles are defined by 4 input parameters:

- the desired number of tiles in the row and column dimensions, respectively N\_TILES\_ROW and N\_TILES\_COL.
- the rate of overlapping area in the row and column dimensions, defined as  $R_OVL_ROW = (L_r g_r)/L_r$  and  $R_OVL_COL = (L_c g_c)/L_c$  with  $0 \le R_OVL_ROW$ ,  $R_OVL_COL \le 0.5$ .

Then  $L_c = COI(N_DET_CAM / (N_TILES_COL.(1-R_OVL_COL) + R_OVL_COL))$  $L_r = COI(N_LINE_OLC / (N_TILES_ROW.(1-R_OVL_ROW) + R_OVL_ROW))$ 



**ISSUE:** 11 Page: 68/107

where COI(x) is the odd integer the closest to x,

and

 $g_c = (N\_DET\_CAM - L_c) / (N\_TILES\_COL - 1)$  $g_r = (N\_LINE\_OLC - L_r) / (N\_TILES\_ROW - 1)$ 

Let the tiles be indexed by the integers (kt, jt), 0  $\leq$  kt  $\leq$  N\_TILES\_ROW-1, 0  $\leq$  jt  $\leq$  N\_TILES\_COL-1, as shown in Figure 3-9

The pixel coordinates of the center of a tile indexed by (kt,jt) are:

$$\label{eq:constraint} \begin{split} jC(jt) &= round(jt.g_c) + (L_c-1)/2 \\ kC(kt) &= round(kt.g_r) + (L_r-1)/2 \end{split}$$

Remark: since  $g_c$  and  $g_r$  may not be integers, the pitch between the centers of adjacent tiles may not be exactly constant. This is not a problem.

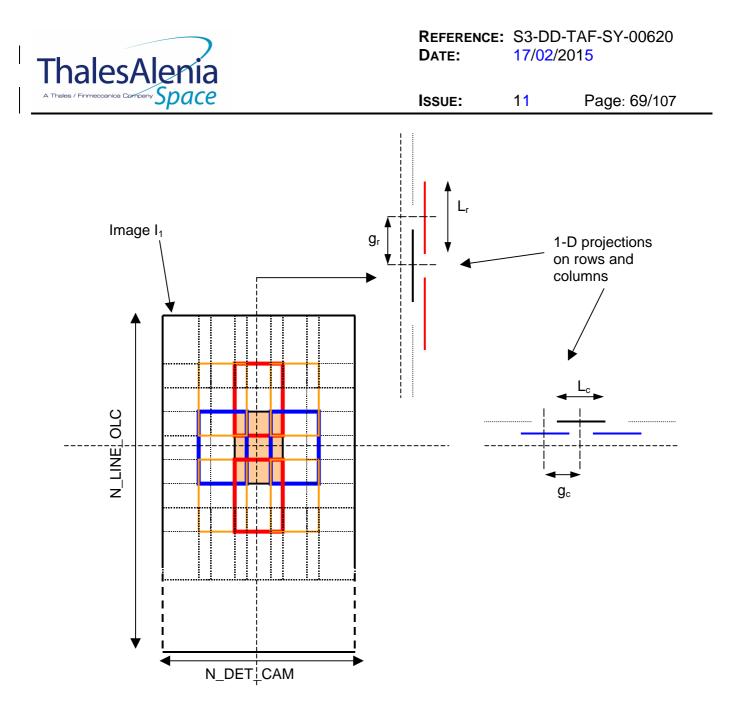


Figure 3-8: The image  $I_1$  is decomposed into overlapping tiles of size  $L_r \propto L_c$  extracted every  $g_r$  pixels in the rows direction and every  $g_c$  pixels in the columns direction.

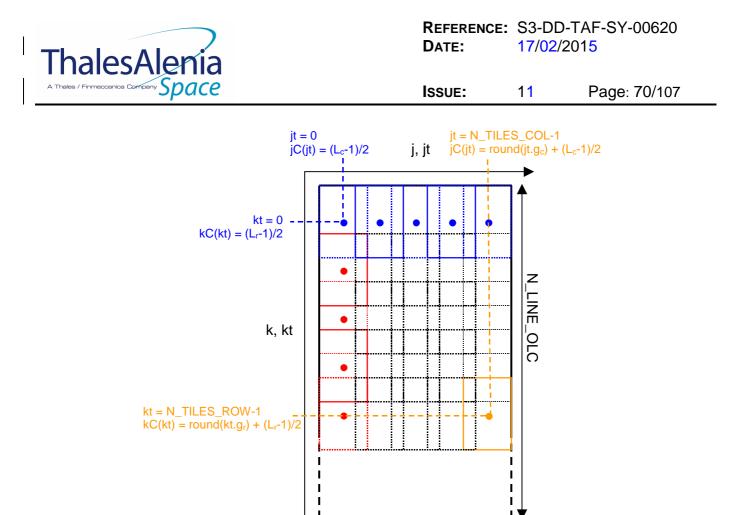


Figure 3-9: Tiles indexing and center coordinates.

N DET CAM

#### Step 2:

For each tile in the image, a unique new shift measurement is created by averaging the shifts measured at tie points in the tile. This averaged shift is associated to a virtual tie-point located at the barycenter of the tie-points in the tile. The algorithm is as follows:

m = 0 // index of the virtual tie-points

For kt = 0 to N\_TILES\_ROW-1

For jt = 0 to N\_TILES\_COL-1

Find the tie-points of the image included in the current tile centered at (kC(jt), jC(kt)). This gives a list of  $N_tp_t$  ile tie-points  $(k_{p_n}, j_{p_n})$  with associated shifts

 $(\hat{\delta}_{p_n}^{row}, \hat{\delta}_{p_n}^{col})$  see section 3.2.4.2.3), n = 1 to *N\_tp\_tile*.

// Note: The number of tie-points per tile shall be stored in an array with the coordinates of the tile center and tile size

If  $N_tp_tile \ge T_N_TP_TILE$  then

// Only tiles containing a certain amount of
// tie-points are processed

m = m + 1

Compute the barycenter  $(\bar{k}_m, \bar{j}_m)$  of the *N\_tp\_tile* tie-points:



**Issue:** 11 Page: 71/107

$$(\bar{k}_m, \bar{j}_m) = (1/N_tp_tile)^{\sum_{n=1}^{N_tp_tile}} (k_{p_n}, j_{p_n})$$

Compute the averaged shifts in the tile:

$$(\overline{\delta}_{m}^{row},\overline{\delta}_{m}^{col}) = (1/N_{tp}tile) \sum_{n=1}^{N_{tr}} (\hat{\delta}_{p_{n}}^{row},\hat{\delta}_{p_{n}}^{col})$$

End if End for jt End for kt

Finally the outputs of this stage are a list of  $N\_virt\_TP$  virtual tie-points  $(\bar{k}_m, \bar{j}_m)$  with the corresponding averaged shifts  $(\bar{\delta}_m^{row}, \bar{\delta}_m^{col})$ , m = 1 to  $N\_virt\_TP$ .

# 3.2.4.3.1.2 Thin-Plate Spline Parameters Estimation

The Thin-Plate Spline (TPS) model belongs to the family of models based on Radial Basis Functions. It is a global model that applies to the whole image I<sub>1</sub>. It is used to approximate a smooth misregistration field  $(\overline{\delta}^{row}(k, j), \overline{\delta}^{col}(k, j))$  from the non-uniformly distributed samples  $(\overline{\delta}^{row}_{m}, \overline{\delta}^{col}_{m})$  located at  $(\overline{k}_{m}, \overline{j}_{m})$ , m = 1 to *N\_virt\_TP*. The two components  $\overline{\delta}^{row}(k, j)$  and  $\overline{\delta}^{col}(k, j)$  are estimated separately by two TPS models which can be written as follows:

$$\overline{\delta}^{row}(k,j) = a_1 + a_2 \cdot k + a_3 \cdot j + \sum_{m=1}^{N_v \cdot irt_m TP} b_m r_m^2 \ln r_m$$
 eq. 3-11

$$\overline{\delta}^{col}(k,j) = e_1 + e_2 \cdot k + e_3 \cdot j + \sum_{m=1}^{N_v - virt_v - TP} f_m r_m^2 \ln r_m$$
 eq. 3-12

with  $r_m^2 = (k - \bar{k}_m)^2 + (j - \bar{j}_m)^2$ .

The parameters are estimated by minimizing a functional criterion (not described here, see [W90] for instance) that includes a "rigidity" parameter  $\lambda \ge 0$ . This comes down to solving the following systems (see [W90]):

$$\begin{cases} (\mathbf{K} + N \_ virt \_ TP.\lambda^{row} \mathbf{I}_{N\_virt\_TP}) \mathbf{B} + \mathbf{MA} = \mathbf{D}^{row} \\ \mathbf{M}^{T} \mathbf{B} = \mathbf{0} \end{cases}$$
eq. 3-13

for the  $\overline{\delta}^{row}(k, j)$  component, and

$$\begin{cases} (\mathbf{K} + N \_ virt \_ TP.\lambda^{col} \mathbf{I}_{N\_virt\_TP})\mathbf{F} + \mathbf{ME} = \mathbf{D}^{col} \\ \mathbf{M}^{T}\mathbf{F} = \mathbf{0} \end{cases}$$
eq. 3-14

for the  $\overline{\delta}^{col}(k, j)$  component,

where:



- **ISSUE:** 11 Page: 72/107
- **I**<sub>N\_virt\_TP</sub> is the identity matrix of size N\_virt\_TP x N\_virt\_TP
- **K** is the *N\_virt\_TP* x *N\_virt\_TP* symmetric matrix such that  $\mathbf{K}_{u,v} = r_{uv}^2 \ln r_{uv}$  with  $r_{uv}^2 = (\bar{k}_u \bar{k}_v)^2 + (\bar{j}_u \bar{j}_v)^2$
- **M** is the *N\_virt\_TP* x 3 matrix such that:

$$\mathbf{M} = \begin{bmatrix} 1 & \bar{k}_{1} & \bar{j}_{1} \\ 1 & \bar{k}_{2} & \bar{j}_{2} \\ \vdots & \vdots & \vdots \\ 1 & \bar{k}_{N_{virt\_TP}} & \bar{j}_{N_{virt\_TP}} \end{bmatrix}$$

• **A**, **B**, **E**, **F** are the vectors of parameters:

$$\mathbf{A} = [a_1, a_2, a_3]^T, \ \mathbf{B} = [b_1, b_2, ..., b_{N_virt_TP}]^T$$
$$\mathbf{E} = [e_1, e_2, e_3]^T, \ \mathbf{F} = [f_1, f_2, ..., f_{N_virt_TP}]^T$$

• **D**<sup>col</sup>, **D**<sup>row</sup> are the vectors of size N\_virt\_TP:

$$\mathbf{D}^{row} = \left[\overline{\delta}_{1}^{row}, \overline{\delta}_{2}^{row}, ..., \overline{\delta}_{N\_virt\_TP}^{row}\right]^{T}$$
$$\mathbf{D}^{col} = \left[\overline{\delta}_{1}^{col}, \overline{\delta}_{2}^{col}, ..., \overline{\delta}_{N\_virt\_TP}^{col}\right]^{T}$$

•  $\lambda^{\text{col}}, \lambda^{\text{row}} \ge 0$  are the rigidity parameters of the TPS of the two components. Their values are given in input:  $\lambda^{\text{col}} = \text{LAMBDA}_{TPS}_{COL}$  and  $\lambda^{\text{row}} = \text{LAMBDA}_{TPS}_{ROW}$ . When  $\lambda^{\text{row}} = 0$  (resp.  $\lambda^{\text{col}} = 0$ ) the corresponding TPS model exactly interpolates the surface at tie points:  $\overline{\delta}^{\text{row}}(\overline{k}_m, \overline{j}_m) = \overline{\delta}_m^{\text{row}}$  (resp.  $\overline{\delta}^{\text{col}}(\overline{k}_m, \overline{j}_m) = \overline{\delta}_m^{\text{col}}$ ). When  $\lambda^{\text{row}} > 0$  (resp.  $\lambda^{\text{col}} > 0$ ), the corresponding TPS is "rigidified" and is an approximation of the surface:  $\overline{\delta}^{\text{row}}(\overline{k}_m, \overline{j}_m) \approx \overline{\delta}_m^{\text{row}}$  (resp.  $\overline{\delta}^{\text{col}}(\overline{k}_m, \overline{j}_m) \approx \overline{\delta}_m^{\text{row}}$ ).

The solution of the systems in eq. 3-13 and eq. 3-14 relies on the QR decomposition of the matrix  $\mathbf{M}$ . This method decomposes the matrix  $\mathbf{M}$  as:

$$\mathbf{M} = \begin{bmatrix} \mathbf{Q}_1, \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where  $[\mathbf{Q}_1, \mathbf{Q}_2]$  is orthogonal and **R** is upper triangular of size 3 x 3.  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  are respectively  $N_{virt}TP \ge 3$  and  $N_{virt}TP \ge 3$  matrices.

Then the solution of the systems in eq. 3-13 and eq. 3-14 is given by the following expressions:

for the  $\overline{\delta}^{row}(k, j)$  component, and



**ISSUE:** 11 Page: 73/107

$$\begin{cases} \mathbf{F} = \mathbf{Q}_{2} (\mathbf{Q}_{2}^{T} \mathbf{K} \mathbf{Q}_{2} + N_{virt_{TP-3}})^{-1} \mathbf{Q}_{2}^{T} \mathbf{D}^{col} \\ \mathbf{E} = \mathbf{R}^{-1} \mathbf{Q}_{1}^{T} (\mathbf{D}^{col} - \mathbf{K} \mathbf{F}) \end{cases}$$
eq. 3-16

for the  $\overline{\delta}^{col}(k, j)$  component.

Note: The QR decomposition method(s) is described in [GV96]. LAPACK is an efficient library of functions for solving the most common problems in numerical linear algebra, including QR decomposition, matrix inversion, etc. It is available in Fortran and C++ languages (LAPACK++, although not as much complete as Fortran LAPACK), respectively at <a href="http://www.netlib.org/lapack/">http://www.netlib.org/lapack/</a> and <a href="http://www.netlib.org/lapack++/">http://www.netlib.org/lapack++/</a>.

# 3.2.4.3.2 Estimation of the Local Deformation Model

The smooth deformation model estimated previously does not take into account local variations of the misregistration field, nor the true density of the tie-points in the image. That is why a local model is needed.

The local deformation model is a piecewise linear model. First the image  $I_1$  is decomposed into pieces (image tessellation) by a Delaunay triangulation of the tie-points. Then the misregistration field is modeled as a 2D linear mapping inside each triangle.

Note: If LOC\_DEF\_MDL\_SWITCH ≠ "YES" then do not compute a local deformation model

# 3.2.4.3.2.1 Delaunay triangulation of the tie points

From a set of disjoint points, the Delaunay triangulation defines a triangular paving, without hole nor overlap, of the convex hull of the points set. The triangles vertices are the points.

The fact that the Delaunay triangulation only covers the convex hull of the point is a drawback of the method for the objective to construct a model that applies to the whole image  $I_1$ . Indeed, the tie-points available in input of the deformation model estimation module will obviously not cover the whole image, particularly its edges (see Figure 3-10).

This problem is overcome by creating additional artificial tie points using the smooth deformation model previously computed. These artificial tie points are regularly distributed everywhere outside the convex hull of the initial tie-points set and especially on the edges of the image (see Figure 3-10). The pitch between two adjacent new tie points is  $\Delta$ \_ATP\_ROW in the row direction and  $\Delta$ \_ATP\_COL in the column direction. These two parameters are given as inputs of the algorithm.

The number of added tie-points is noted *N\_add\_TP* and these tie-points  $(k_{p'}, j_{p'})$  are indexed by  $p' = 1, ..., N_add_TP$ . The misregistration values at these tie-points are obtained applying the smooth deformation model to the  $(k_{p'}, j_{p'})$ :  $(\overline{\delta}^{row}(k_{p'}, j_{p'}), \overline{\delta}^{col}(k_{p'}, j_{p'}))$ .

The complete list of tie-points is

 $\{(k_p, j_p), p = 1, ..., N_TP_L lC_END(m)\} \cup \{(k_{p'}, j_{p'}), p' = 1, ..., N_add_TP\}$ . The tie points in this list



are indifferently indexed by the integer p" = 1,...,(N\_TP\_L1C\_END(m) + N\_add\_TP). The corresponding (smooth or true) misregistration shift is noted  $(\tilde{\delta}_{p^{"}}^{row}, \tilde{\delta}_{p^{"}}^{col})$ .

Then the Delaunay triangulation function is fed with the complete list of (true and additional) tiepoints. The output is a list of  $N_TRI$  triangles, represented by their 3 vertices.

Remark: The 2D Delaunay triangulation is a classical tool in computer graphics, and many efficient and widely validated libraries already exist (see for instance the qdelaunay function in the Qhull package at <u>http://www.qhull.org/</u>). For this reason the algorithms are not described here.

Thus the Delaunay triangulation function takes a set of tie-point coordinates in input and outputs a set of triangles represented by their 3 vertices (3 tie-points).

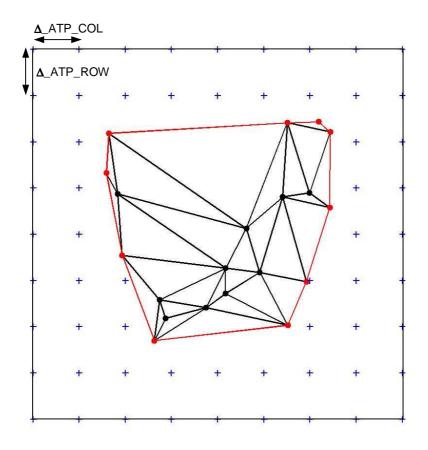


Figure 3-10: Example of a Delaunay triangulation with 20 tie-points (red and black points). The convex hull of the tie-points set is shown in red. The image  $I_1$  is the black box. The blue crosses are additional artificial tie-points, computed by means of the smooth model and used to build a Delaunay triangulation on the whole image (see Figure 3-11).



**Issue:** 11 Page: 75/107

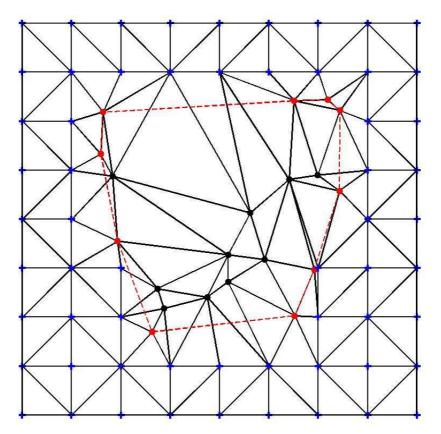


Figure 3-11: Delaunay triangulation of the full set of tie-points, including the initial true tie-points (red and black points) and the additional artificial tie-points (blue crosses) computed by means of the smooth model. The convex hull of the initial tie-points set is shown in red dotted line.

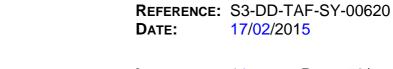
# 3.2.4.3.2.2 Linear model estimation in each triangle

For each triangle in the list returned by the Delaunay triangulation, perform the following operations:

Let  $(k_{p_1^r}, j_{p_1^r})$ ,  $(k_{p_2^r}, j_{p_2^r})$  and  $(k_{p_3^r}, j_{p_3^r})$  be the 3 vertices of the current triangle. In this triangle, the linear model has the form:

$$\delta^{row}(k,j) = P_{\alpha}(k,j) = \alpha_1 + \alpha_2 k + \alpha_3 j \qquad \text{eq. 3.1}$$

$$\delta^{col}(k, j) = P_{\beta}(k, j) = \beta_1 + \beta_2 k + \beta_3 j$$
 eq. 3.2



**ISSUE:** 11 Page: 76/107

Imposing that this model is equal to the misregistration shifts at each vertex ( $\tilde{\delta}_{p_i^*}^{row} = P_{\alpha}(k_{p_i^*}, j_{p_i^*})$  and  $\tilde{\delta}_{p_i^*}^{col} = P_{\beta}(k_{p_i^*}, j_{p_i^*})$  for i=1,2,3) leads to the following systems of equations, with  $\alpha$ 's and  $\beta$ 's

and  $\sigma_{p_i^{\prime}} = \Gamma_{\beta}(\kappa_{p_i^{\prime}}, J_{p_i^{\prime}})$  for i=1,2,3) leads to the following systems of equations, with  $\alpha$ 's and  $\beta$ 's as unknowns:

$\left[ \begin{array}{c} \widetilde{\delta}_{p_1^{\prime}}^{row} \\ \widetilde{\delta}_{p_2^{\prime}}^{row} \\ \widetilde{\delta}_{p_3^{\prime}}^{row} \end{array} \right]$	$= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$k_{p_1''} \ k_{p_2''} \ k_{p_3''}$	$j_{p_1''} \ j_{p_1''} \ j_{p_1''}$	$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$
$egin{bmatrix} \widetilde{\delta}_{p_1^{\prime}}^{col}\ \widetilde{\delta}_{p_2^{\prime}}^{col}\ \widetilde{\delta}_{p_3^{\prime}}^{col} \end{bmatrix}$	$=\begin{bmatrix}1\\1\\1\end{bmatrix}$	$k_{p_1''} \ k_{p_2''} \ k_{p_3''}$	$egin{array}{c} j_{p_1''} \ j_{p_1''} \ j_{p_1''} \ j_{p_1''} \end{array}$	$egin{bmatrix} eta_1\ eta_2\ eta_3 \end{bmatrix}$

and

or in a shorter evident matrix notation,

alesAle

 $\mathbf{D}^{\text{col}} = \mathbf{W}\boldsymbol{\beta}$ The solutions of these systems are straightforward, relying on the inversion of a simple 3x3 matrix W:

$$\boldsymbol{\alpha} = \mathbf{W}^{-1}\mathbf{D}^{\text{row}}$$
$$\boldsymbol{\beta} = \mathbf{W}^{-1}\mathbf{D}^{\text{col}}$$

 $\mathbf{D}^{\mathrm{row}} = \mathbf{W}\boldsymbol{\alpha}$ 

End for each triangle

The output of this processing is a set of  $\alpha_u$  and  $\beta_u$  (vectors  $\alpha$  and  $\beta$ ) model parameters for each triangle. There is no a priori reason for these parameters to be identical from a triangle to another.

3.2.4.3.3 Computation of the deformation field on  $I_1$ 

Note: If LOC\_DEF\_MDL\_SWITCH  $\neq$  "YES" then this stage is replaced by computation of the deformation field at every pixel of I<sub>1</sub>using the Smooth deformation model (eq. 3-11 and eq. 3-12)

The piecewise linear deformation model is now used to compute the misregistration shifts at each pixel of the image  $I_1$ , forming a deformation field (or grid)

For each pixel (k,j) in the I<sub>1</sub> image (k = 0,...,N\_LINE\_OLC-1 and j = 0,..., N\_DET\_CAM-1)

Find in which triangle (k,j) falls. Remark: If (k,j) is on an edge separating two triangles, this is not a problem since the linear models of two adjacent triangles coincide on their common edge.

Compute the misregistration shift  $(\hat{\delta}^{row}(k, j), \hat{\delta}^{col}(k, j))$ , using the linear models corresponding to the found triangle (eq. 3.1 and eq. 3.2)

End for each pixel

ThalesAlenia		REFERENCE:         S3-DD-TAF-SY-00620           DATE:         17/02/2015	
A Thales / Finmeccanica Company Space	ISSUE:	11	Page: 77/107

Remark: The piecewise linear deformation model is continuous. Unfortunately, its derivative is not continuous on the edges of the triangles. If this discontinuity turns out to be problematic, a dedicated processing should be performed to smooth the deformation field obtained as described above.

The estimated piece-wise correspondence grid between image  $I_1$  and Image  $I_2$  projected to image  $I_1$  is now noted

$$(\mathbf{k}_{2}, \mathbf{j}_{2}) = \hat{\mathbf{D}}_{12}(\mathbf{k}, \mathbf{j}) = (\mathbf{k}, \mathbf{j}) + (\hat{\delta}^{row}(k, \mathbf{j}), \hat{\delta}^{col}(k, \mathbf{j}))$$

with (k, j) and  $(k_2, j_2)$  the coordinates in image  $I_1$  and Image  $I_2$  projected to image  $I_1$  respectively.  $(\hat{\delta}^{row}(k, j), \hat{\delta}^{col}(k, j))$  is the modeled misregistration error at point (k, j) in image  $I_1$ .

This model is designed to have good performances only for pixels over land (which are at a reasonable distance of tie points). To avoid interpolation problems in area far from tie-points the variation range of the deformation model is limited:

For each (k,j) do

If 
$$\sqrt{\hat{\delta}^{row}(k,j)^2 + \hat{\delta}^{col}(k,j)^2}$$
 > MAX\_DELTA\_EST then set  $\hat{\delta}^{row}(k,j) = \hat{\delta}^{col}(k,j) = 0$ 

End for

The estimated correspondence grid between image  $I_1$  and Image  $I_2$  *in their acquisition geometry* is got applying the erroneous correspondence model  $G_{12}$  to the correspondence grid in image  $I_1$  geometry:

$$(k', j') = \hat{G}_{12}(k, j) = G_{12} \circ \hat{D}_{12}(k, j)$$
 eq. 3-17

with (k,j) and (k',j') coordinates in image  $\mathsf{I}_1$  and Image  $\mathsf{I}_2$  in their acquisition geometry respectively.

The algorithm is as follows:

For each pixel (k,j) in the  $I_1$  image do

Read  $(k_2, j_2) = \hat{D}_{12}(k, j)$  (non-integer coordinates)

Compute the ortho-rectified geolocation of  $(k_2, j_2)$  by interpolation in the ortho-rectified geolocation grid of image I<sub>1</sub> (see Annex A):  $(\lambda, \phi) = \text{Loc}_1(k_2, j_2)$ 

Compute  $(k',j') = Loc_2^{-1}(\lambda, \phi)$  the inverse geolocation of  $(\lambda, \phi)$  in image  $I_2$  (see Annex A). // The ortho-rectified geolocation grid of image  $I_2$  is passed to the inverse geolocation function. See below for the initialization of the function.

If (k',j') is outside the SLSTR image (k'<0 OR k' $\geq$ N\_SCAN\_SLST\_NAD\_05km\_CUT OR j'<0 OR j' $\geq$ N\_PIX\_SCAN\_NAD\_05km) then set (k',j') = (NO\_CORRESP\_VAL, NO\_CORRESP\_VAL)

End For

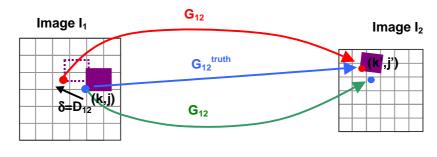
Initialization of the inverse geolocation function:

If the current pixel processed is the first pixel of the list (for instance the central pixel of the image I<sub>1</sub>) then find the nearest neighbor of (λ, φ) in the ortho-rectified geolocation grid of image I<sub>2</sub> (SLSTR S<sub>u</sub> channel) and consider the corresponding (k',j') coordinates in this grid as a first guess for the inverse geolocation solution.



For any other pixel in the list the corresponding (k',j') first guess is the result of the correspondence function previously computed for a neighboring pixel (k<sub>old</sub>, j<sub>old</sub>) in the image I<sub>1</sub>. This implies that the pixels (k,j) in image I<sub>1</sub> must be processed gradually, from one pixel to its neighbor(s). Figure 3-4 shows an example of pixels processing order in the Search imagette that can be adapted to the whole image I<sub>1</sub>.

Figure 3-12 illustrates the relation between the different (erroneous and true) correspondence functions.





Superposition of image  $I_2$  (in image  $I_1$  geometry) on image  $I_1$ 

Figure 3-12: Retrieving the true correspondence function  $G_{12}^{truth}$  between Image 1 and image 2 from the function  $G_{12}$  and misregistration information  $\delta$  (or  $D_{12}$ ).  $G_{12}^{truth}(k,j) = G_{12}[(k,j) + D_{12}(k,j)]$ .

# $3.3\,$ Computing the correspondence between OLCI Oq pixels and all other SLSTR and OLCI channels

# 3.3.1 Algorithm Inputs

# 3.3.1.1 OLCI inter-channel spatial misregistration Auxiliary Data File

This datafile, obtained from characterization data of OLCI instrument contains inter-channel spatial misregistration between the OLCI  $O_q^m$  reference channel and the other channels  $O_b^m$ ,  $b \neq q$  in each camera module m: the shift vectors  $(d_m^{row,b}(j), d_m^{col,b}(j))$ , independent of the time (i.e. independent of k), between each OLCI  $O_q^m$  detector j and the same detector in another channel  $O_b^m$ . Thus, in the camera module m, the correspondence models are  $(k^b, j^b) = T_{q,b}^m(k, j) = (k,j) + (d_m^{row,b}(j), d_m^{col,b}(j))$ .

The format and contents of this auxiliary data file are described in [RD-5]. The  $d_m^{row,b}(j)$  and  $d_m^{col,b}(j)$  correspond to the array Row\_Shift and Col\_Shift respectively.

# 3.3.1.2 SLSTR nadir view inter-channel spatial misregistration Auxiliary Data File

This Auxiliary Data File, obtained from characterization data of SLSTR instrument, contains static inter-channel spatial misregistration, given for one scan, between the SLSTR  $S_u$  reference channel and the other SLSTR  $S_{b'}$  channels (b' = 1,...11, b' $\neq$ u): for all ( $k_{det}^u$ ,j) in one scan of  $S_u$ 



 $(k_{det}^{u} = 0 \text{ to } 3 \text{ and } j = 0 \text{ to } N_PIX_SCAN_NAD_05km-1 \text{ it gives the corresponding locations}$  $(k_{det}^{b'}, j^{b'})$  in the scan of the S<sub>b'</sub> channels, b' $\neq$ u:  $k_{det}^{b'} = row_corresp[b'](k_{det}^{u}, j)$  and  $j^{b'} = col_corresp[b'](k_{det}^{u}, j)$ 

The format and contents of this auxiliary data file are described in [RD-5].

Note: for each SWIR channel S4, S5, S6, there must be 3 correspondence grids available (for "A", "B" and TDI sub-bands). For each SWIR channel suited grid must be selected according to the chosen sub-bands ("A", "B" or "averaged") defined by the SLST\_SWIR\_SELECT input parameter.

# 3.3.1.3 SLSTR $S_u$ (nadir view) / OLCI $O_q^m$ inter-instruments misregistration grids

These are the models established in section 3.2.4.3 (eq. 3-17), one for each camera module image of OLCI. The deformation grids between SLSTR  $S_u$  and OLCI  $O_q^m$  in their acquisition geometry (eq. 3-17) are now noted  $(k',j') = R^m(k,j)$  for each coordinates (k,j) in OLCI  $O_q^m$  image m, and (k',j') in SLSTR  $S_u$ ,  $k = 0,...,N_LINE_OLC-1$ ,  $j = 0,...,N_DET_CAM-1$ .

# 3.3.1.4 OLCI $O_{q}^{m}$ ortho-rectified geolocation grids

Extracted from OLCI L1b product by L1c pre-processing (paragraph 3.1)

3.3.1.5 SLSTR oblique view ortho-rectified geolocation grids for 500m and 1km channels

Extracted from SLSTR L1b product by L1c pre-processing (paragraph 3.1.4.3.2.3)

# 3.3.1.6 Processing Parameters

- Interpolation parameters (bicubic or bilinear interpolation)
- Parameters for the inverse ortho-rectified geolocation function (see section A.2.1)
- NO\_CORRESP\_VAL
- SLST\_1km\_K\_MARGIN (even number)
- SLST\_1km\_J\_MARGIN (even number)
- SLST\_SWIR\_SELECT
- COLLOC\_NN\_RADIUS

# 3.3.2 Processing Objective

The objective of this stage is to:

- compute the correspondence between the OLCI reference channel O<sub>q</sub><sup>m</sup> in its acquisition geometry (m=1,...,5) and all the other OLCI and SLSTR nadir view channels (SLSTR S<sub>b</sub><sup>,</sup> with b' = 1...11, OLCI O<sub>b</sub><sup>m</sup> with b = 1...21, b≠q).
- compute the correspondence between the OLCI reference channel O<sub>q</sub><sup>m</sup> (m=1,...,5) and all the SLSTR oblique view channels in their acquisition geometry. This processing establishes a *collocation* grid between OLCI and SLSTR oblique view images, based on their superimposition in the gridded L1b product. Note that the SLSTR oblique view is not subjected to any co-registration performance requirement, as agreed by ESA.



 Cut the SLSTR nadir and oblique view images and annotation datasets close to the OLCI image borders

The correspondence between a pixel (k,j) of the  $O_q^m$  channel and the channel  $O_b^m$  is directly given in input (section 3.3.1.1) while the correspondence between  $O_q^m$  and the SLSTR channels needs to be established by processing.

These correspondence data will be included in the L1c product. Figure 3-13 shows the correspondence relations between the different channels.

# 3.3.3 Algorithm Outputs

The main outputs of this module are

- the 5 x 31 misregistration correspondence grids (m=1,...,5) between the pixels (k,j) of the OLCI O<sub>q</sub><sup>m</sup> channel and its sub-pixel (k',j') locations in the other OLCI O<sub>b</sub><sup>m</sup> (b≠q) and SLSTR nadir view S<sub>b'</sub> channels.
- The 5 x (1 + n\_sub\_bands) collocation grids (m=2,...,5) between the pixels (k,j) of the OLCI O<sub>q</sub><sup>m</sup> channel and its sub-pixel (k',j') corresponding location in the SLSTR oblique view 500m and 1km channels. n\_sub\_bands = 1 to 3 depending on the number of sub-bands ("A", "B" and TDI) involved in the 500m channels. In baseline the TDI sub-band is selected for all SWIR channel, while the Visible channels are defined as "A" sub-bands, therefore, n\_sub\_bands = 2. All 500m channels acquired by a same sub-band are considered as one (i.e. perfectly collocated). Similarly, all 1km channels are considered as one. Only the common swath of the two images must be processed, hence the common swath has to be determined first.
- All the SLSTR channels and related annotations cut close to the common swath (plus margins), with offsets allowing to retrieve the original instrument scan numbers and relative pixel numbers:
  - L1C\_SCAN\_OFFSET\_ALT\_1km, L1C\_PIX\_OFFSET\_ALT\_1km,
  - L1C\_SCAN\_OFFSET\_ALT\_05km, L1C\_PIX\_OFFSET\_ALT\_05km,
  - L1C\_SCAN\_OFFSET\_NAD\_1km, L1C\_PIX\_OFFSET\_NAD\_1km,
  - L1C\_SCAN\_OFFSET\_NAD\_05km, L1C\_PIX\_OFFSET\_NAD\_05km.

These data will be included in the L1c product.

The above grids shall be output as intermediate verification/analysis data by the Processor



**Issue:** 11 Page: 81/107

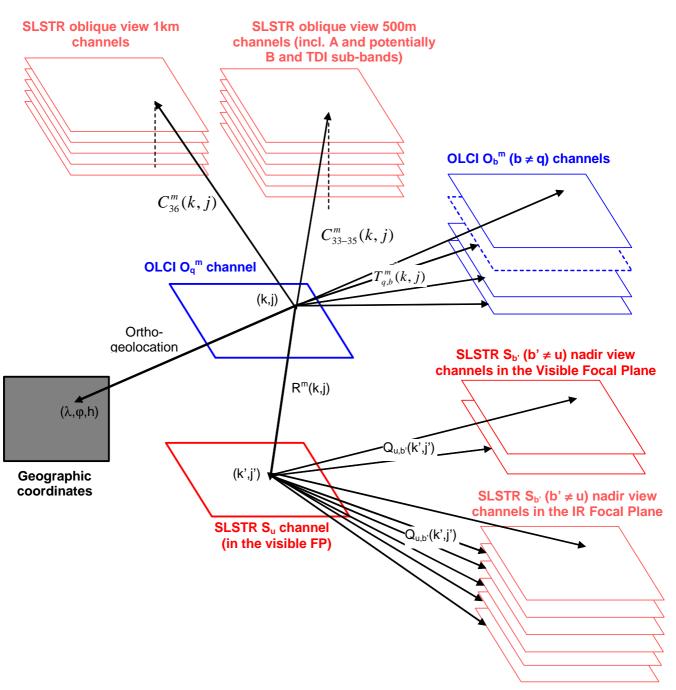


Figure 3-13: Correspondence relations between the OLCI O<sub>q</sub><sup>m</sup> reference channel and the other OLCI, SLSTR nadir and oblique view channels (applicable for m=1,...,5). The reference ortho-rectified geolocation information included in the L1c product is also represented.



# 3.3.4 Mathematical Description

The L1c image includes 43 bands, numbered from 1 to 43, corresponding to OLCI and SLSTR bands and ordered as  $O_1$ ,  $O_2$ ,...,  $O_{21}$ ,  $S_1$ ,  $S_2$ ,...,  $S_{11}$ ,  $S_1^{oblique}$ ,  $S_2^{oblique}$ ,...,  $S_{11}^{oblique}$ . They are noted  $B_d$ , d = 1...43. Remark:  $S_{10}$  and  $S_{11}$  (resp.  $S_{10}^{oblique}$  and  $S_{11}^{oblique}$ ) stand for the nadir (resp. oblique) view F1 and F2 channels.

We now establish the correspondence grids between each pixel (k,j) of the OLCI  $O_q^m$  reference channel (m=1,...,5) and any of the OLCI or SLSTR nadir and oblique view channels in their acquisition geometry. These correspondence grids are noted  $(k^d,j^d) = C_d^m(k,j)$ , where (k,j) and  $(k^d,j^d)$  are respectively the  $O_q^m$  image coordinates and the coordinates in the band B<sub>d</sub> in their acquisition geometry. Since the oblique view 500m (resp. 1km) channels acquired with the same sub-band are considered as a single channel for collocation there is only one correspondence grid per camera module for the collocation of  $O_q^m$  and the SLSTR oblique view 500m (resp. 1km) channels acquired with the same sub-band. Hence:

- for the nadir view correspondence grids, d=22,...,32, d≠q
- for the oblique view correspondence grids:
  - $\circ$  d = 33 refers to the grid of 500m A sub-bands
    - $\circ$  d = 34 refers to the grid of 500m B sub-bands (if any)
    - $\circ$  d = 35 refers to the grid of 500m TDI sub-bands (if any)
    - $\circ$  d = 36 refers to the grid of the 1km channel.

At the end of this processing all the grids  $C_d^m(k, j)$  are written into the level 1c product, except for d = 34 and 35 that are written only if any B or TDI sub-bands are specified by the SLST\_SWIR\_SELECT input parameter.

# 3.3.4.1 Establishing the correspondence between OLCI $O_q^m$ and OLCI $O_b^m$ channels (b $\neq q$ )

In baseline, for storage efficiency, this information is a copy of the OLCI inter-channel spatial misregistration Auxiliary Data File, that is local shifts  $(d_m^{row,b}(j), d_m^{col,b}(j))$  between  $O_q^m$  channel and the channel  $B_d^m$  for d = 1,...,21,  $d \neq q$  (or  $O_b^m$  for b = 1,...,21,  $b \neq q$ ).

Nevertheless the O-GPP shall be able to construct and store 2D deformation models such as  $(k^{b}, j^{b}) = T_{a,b}^{m}(k, j) = (k,j) + (d_{m}^{row,b}(j), d_{m}^{col,b}(j)).$ 

From now on, the 1D or 2D correspondence model is indistinctly noted, for m=1,...,5, for d = 1,...,21, d $\neq$ q:

$$(\mathbf{k}^{d}, \mathbf{j}^{d}) = C_{d}^{m}(k, j) = T_{q,d}^{m}(k, j)$$
 eq. 3-18

with  $(k^d,j^d)$  the sub-pixel corresponding location in the product  $B_d^m$  (or OLCI  $O_b^m$ ) in its acquisition geometry.

# 3.3.4.2 Establishing the correspondence between OLCI O<sub>a</sub><sup>m</sup> and SLSTR S<sub>u</sub> channel

The correspondence between a pixel (k,j) of the  $O_q^m$  channel and the channel  $B_{21+u}$  (or SLSTR  $S_u$  channel) is directly given in input (section 3.3.1). For m=1,...5:

$$(k^{21+u}, j^{21+u}) = C_{21+u}^m(k, j) = R^m(k, j)$$
 eq. 3-19

ThalesAlenia	REFERENCE:         S3-DD-TAF-SY-00620           DATE:         17/02/2015		
A Thales / Finmeccanica Company Space	ISSUE:	11	Page: 83/107

with  $(k^{21+u}, j^{21+u})$  the sub-pixel corresponding location in the product band  $B_{21+u}$  (or SLSTR  $S_u$  channel) in its acquisition geometry.

# 3.3.4.3 Establishing the correspondence between OLCI $O_q^m$ and SLSTR $S_{b'}$ channels (b' $\neq$ u)

The correspondence between a pixel (k,j) of the  $O_q^m$  channel and the channel  $B_d$  for d=22,...,32,  $d \neq 21+u$  (or SLSTR  $S_{b'}$  channel for b'=1,...11, b' $\neq u$ ), is, for m=1,...,5:

 $(k^{d},j^{d}) = C_{d}^{m}(k,j) = Q_{u,d-21} \circ R^{m}(k,j) = Q_{u,d-21}(k^{21+u},j^{21+u})$  eq. 3-20

where  $(k^{b'},j^{b'}) = Q_{u,b'}(k^{u},j^{u})$  is the correspondence grid between SLSTR  $S_u$  channel and  $S_{b'}$  channel, and with  $(k^d,j^d)$  the sub-pixel corresponding location in the product band  $B_d$  (or SLSTR  $S_{b'}$  channel) in its acquisition geometry.

 $(k^{d},j^{d})$  is computed by bicubic interpolation in the  $Q_{u,d-21}$  grid given as input, at location  $(k^{21+u},j^{21+u}) = C_{21+u}^{m}(k,j) = \mathbb{R}^{m}(k,j)$  (see Figure 3-14).

The algorithm is as follows (for a given camera module m):

For each (k,j) in the  $O_q^m$  channel do

Read the corresponding sub-pixel location  $(k^u, j^u)$  in the SLSTR S<sub>u</sub> channel in the R<sup>m</sup>(k,j) grid given in input

Compute  $k_{det}^{u} = k^{u} \mod 4$  // the (**non-integer**) remainder of the division of k<sup>u</sup> by 4

Compute S =  $(k^{u} - k_{det}^{u}) / 4$  // the quotient of division of k<sup>u</sup> by 4

For each SLSTR channel  $S_{b'}$  (b'  $\neq$  u) do

Compute  $k_{det}^{b'} = row\_corresp[b'](k_{det}^{u}, j^{u})$  by bicubic interpolation in the table  $row\_corresp[b'](k_{det}^{u}, j^{u})$ 

Compute  $j^{b'} = col\_corresp[b'](k_{det}^{u}, j^{u})$  by bicubic interpolation in the table  $col\_corresp[b'](k_{det}^{u}, j)$ 

If b' <= 6 then // 500m channels  $k^{b'} = 4.S + k_{det}^{b'}$ End If If b' >= 7 then // 1km channels  $k^{b'} = 2.S + k_{det}^{b'}$ End If Finally,  $C_{21+b'}^{m}(k, j) = (k^{b'}, j^{b'})$ 

End For

End For



**ISSUE:** 11 Page: 84/107

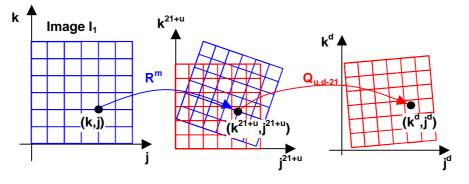


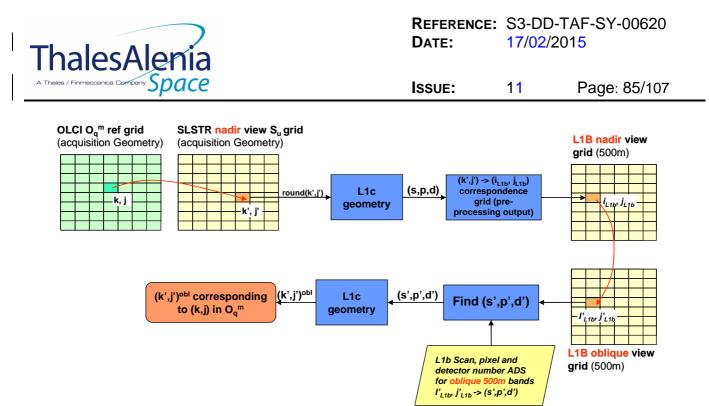
Figure 3-14: Representation of  $(k^d, j^d) = C_d^m(k, j)$ , for d=22,...,32, d  $\neq$  21+u. The grid  $R^m$  (in blue) allows retrieving  $(k^{21+u}, j^{21+u})$  in the SLSTR S<sub>u</sub> channel from (k,j) in the OLCI O<sub>q</sub><sup>m</sup> channel (without interpolation since the (k,j) are integers). Then the input tables *row\_corresp* and *col\_corresp* allows retrieving  $(k^d, j^d)$  in the SLST S<sub>d-21</sub> channel from  $(k^{21+u}, j^{21+u})$ , with interpolation since the found  $(k^{21+u}, j^{21+u})$  coordinates are possibly not integers.

# 3.3.4.4 Establishing the collocation grids between OLCI O<sub>g</sub><sup>m</sup> and SLSTR oblique view channels

The computation of the collocation grids between OLCI  $O_q^m$  and the SLSTR oblique view channels is based on SLSTR L1b gridded image. Indeed in SLSTR L1b product, the image of the nadir and oblique views are already "collocated" on the L1b image grid. Thus principle of the collocation algorithm is as follows:

- 1. For a given OLCI pixel in O<sub>q</sub><sup>m</sup>, the corresponding location in SLSTR nadir view S<sub>u</sub> channel in acquisition geometry is already known in R<sup>m</sup> (k,j) (see paragraph 3.3.4.2)
- 2. Then the location of this pixel in the SLSTR L1b image can be retrieved using the (k',j') -> (i<sub>L1b</sub>, j<sub>L1b</sub>) tables build during pre-processing (section 3.1.4.3.2.2.1)
- 3. The L1b image coordinates of the corresponding oblique view pixel is found using superimposition of nadir and oblique views images on L1b product grid
- 4. Finally the image coordinates of the oblique view pixel is converted to L1c acquisition geometry coordinates using L1b product information as in pre-processing stage

This is illustrated in Figure 3-15.



# Figure 3-15: Principle of the algorithm for SLSTR oblique view / OLCI reference channel collocation

As stated above the collocation grids are not computed for each SLSTR channel but for each SLSTR sub-band (A, B, TDI or 1km), considering that the channels in a given sub-band have the same collocation.

Thus, the number of collocation grids to be computed depends on the SWIR sub-bands ("A", "B" or "TDI") included in the L1c product (specified by the SLST\_SWIR\_SELECT processing parameter). The 2 grids for the "A" sub-band and for the 1km channels are always present. But if there are "B" or TDI SWIR sub-bands included in the L1c product then corresponding collocation grids must be computed accordingly.

# 3.3.4.4.1 Pseudo-code of the collocation algorithm

From the viewing geometry characteristics of OLCI and SLSTR oblique view,

- The swath of the OLCI camera module 1 is outside the SLSTR oblique view swath, thus is not considered. However "dummy" collocation grids  $C_{\#\#}^1(k, j)$  are included in the product, where ## is to be replaced by 33 to 35 depending on the selected A, B or TDI sub-bands.
- The swath of the OLCI camera module 2 is only partially included in the SLSTR oblique view swath
- The swaths of the OLCI camera modules 3, 4 and 5 are included in the SLSTR oblique view swath

Read useful L1b parameters in SLST L1b product:

- Position of element (0,\*) wrt. L1b product start for nadir view S<sub>u</sub> channel and oblique view sub-bands:
  - Start\_Offset\_Su\_Nad = Read start\_offset global attribute in Scan, pixel and detector number ADS of nadir view "A" stripe
  - Start\_Offset\_A\_Obl = Read start\_offset global attribute in Scan, pixel and detector number ADS of oblique view "A" stripe



**ISSUE:** 11 Page: 86/107

- Start\_Offset\_B\_Obl = Read start\_offset global attribute in Scan, pixel and detector number ADS of oblique view "B" stripe (only if necessary)
- Start\_Offset\_TDI\_Obl = Read start\_offset global attribute in Scan, pixel and detector number ADS of oblique view "TDI" stripe (only if necessary)
- Start\_Offset\_1km\_Obl = Read start\_offset global attribute in Scan, pixel and detector number ADS of oblique view 1km channel
- Position of element (\*,0) wrt sub-satellite track for nadir view S<sub>u</sub> channel and oblique view sub-bands:
  - Track\_Offset\_Su\_Nad = Read track\_offset global attribute in Scan, pixel and detector number ADS of nadir view "A" stripe
  - Track\_Offset\_A\_Obl = Read track\_offset global attribute in Scan, pixel and detector number ADS of oblique view "A" stripe
  - Track\_Offset\_B\_Obl = Read track\_offset global attribute in Scan, pixel and detector number ADS of oblique view "B" stripe (only if necessary)
  - Track\_Offset\_TDI\_ObI = Read track\_offset global attribute in Scan, pixel and detector number ADS of oblique view "TDI" stripe (only if necessary)
  - Track\_Offset\_1km\_Obl = Read track\_offset global attribute in Scan, pixel and detector number ADS of oblique view 1km channels

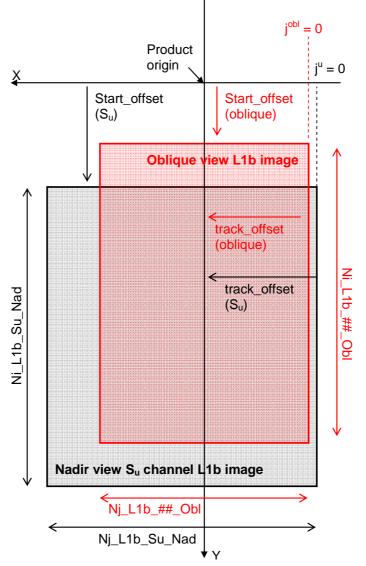
# • SLST L1b image dimensions

Warning: in the L1c ATBD we adopt the same image index convention (i,j) as in SLSTR ATBD 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). However in the SY-4 product definition the convention is opposite.

- Ni\_L1b\_Su\_Nad = Read along-track array size of the Scan, pixel and detector number ADS of nadir view "A" stripe (nj parameter)
- Ni\_L1b\_A\_Obl = Read along-track array size of the Scan, pixel and detector number ADS of oblique view "A" stripe (nj parameter)
- Ni\_L1b\_B\_Obl = Read along-track array size of the Scan, pixel and detector number ADS of oblique view "B" stripe (nj parameter) (if necessary)
- Ni\_L1b\_TDI\_Obl = Read along-track array size of the Scan, pixel and detector number ADS of oblique view "TDI" stripe (nj parameter) (if necessary)
- Ni\_L1b\_1km\_Obl = Read along-track array size of the Scan, pixel and detector number ADS of oblique view 1km channel (nj parameter)
- Nj\_L1b\_Su\_Nad = Read across-track array size of the Scan, pixel and detector number ADS of nadir view "A" stripe (ni parameter)
- Nj\_L1b\_A\_Obl = Read across -track array size of the Scan, pixel and detector number ADS of oblique view "A" stripe (ni parameter)
- Nj\_L1b\_B\_Obl = Read across -track array size of the Scan, pixel and detector number ADS of oblique view "B" stripe (ni parameter) (if necessary)
- Nj\_L1b\_TDI\_Obl = Read across -track array size of the Scan, pixel and detector number ADS of oblique view "TDI" stripe (ni parameter) (if necessary)
- Nj\_L1b\_1km\_Obl = Read across -track array size of the Scan, pixel and detector number ADS of oblique view 1km channel (ni parameter)



These parameters allow retrieving position of L1b oblique view images relative to the nadir view  $S_u$  image as shown in Figure 3-16.





Initialize the correspondence (collocation) grids:

 $C_b^m(k, j) = (NO_CORRESP_VAL, NO_CORRESP_VAL)$ , for m=1 to 5, k=0 to N\_LINE\_OLC-1, j=0 to N\_DET\_CAM-1, b=33, 36 and 34, 35 if necessary

For OLCI camera module m = 2 to 5 For all (k,j), k=0 to N\_LINE\_OLC-1, j=0 to N\_DET\_CAM-1 Read (k', j') =  $R^m(k,j)$  // the corresponding location in SLSTR S<sub>u</sub> channel k\_r' = round(k') j\_r' = round(j') Find back the corresponding location in the L1b nadir view gridded image using tables built during pre-processing



 $i_{I1b}^{Su} = i\_L1b\_Sref(k\_r',j\_r')$  $j_{L1b}^{Su} = j_L1b_Sref(k_r',j_r')$ If  $(i_{L1b}^{Su}, j_{L1b}^{Su}) \neq (NO\_CORRESP\_VAL, NO\_CORRESP\_VAL)$ For \$\$ = A, 1km, and possibly B, TDI (depending on SLST\_SWIR\_SELECT parameter) Find the corresponding SLSTR oblique view location in L1b gridded \$\$ stripe: If \$ = A, B or TDI  $i_{L1b}^{Obl_{\$\$}} = i_{L1b}^{Su} + Start_Offset_Su_Nad - Start_Offset_{\$_Obl}$  $j_{L1b}^{Obl_{\$}} = j_{L1b}^{Su}$  - Track\_Offset\_Su\_Nad + Track\_Offset\_\$\$\_Obl End If If \$\$ = 1km  $i_{L1b}^{Obl\_\$\$} = \mathsf{floor}((i_{L1b}^{Su} + Start\_Offset\_Su\_Nad)/2) - Start\_Offset\_\$\$\_Obl$  $j_{Llb}^{Obl}$  = floor((  $j_{Llb}^{Su}$  - Track\_Offset\_Su\_Nad )/2) + Track\_Offset\_\$\$\_Obl End If Check if found location is in the L1b gridded \$\$ stripe of oblique view: If  $i_{L1b}^{Obl-\$\$} \ge 0$  and  $i_{L1b}^{Obl-\$\$} < Ni_L1b_\$\$_Obl$  and  $j_{L1b}^{Obl-\$\$} \ge 0$  and  $j_{L1b}^{Obl-\$\$} < 0$ Nj\_L1b\_\$\$\_Obl Then Read the scan number S, (relative) pixel number pa', detector number d of pixel  $(i_{L1b}^{Obl}, j_{L1b}^{Obl})^5$  from the Scan, pixel and detector number ADS of oblique view A sub-band in SLST L1b product Compute the instrument scan number: s = scale\*S + d (scale = 4 for \$\$ = A, B or TDI, scale = 2 for \$\$ = 1km)  $C_b^m(k, j) = (s - s_{\min}^{a,05km}, p_a')$  if \$\$ = A, B or TDI with b=33,34,35 respectively  $C_{36}^{m}(k, j) = (s - s_{\min}^{a, 1km}, p_{a}')$  if \$\$ = 1km End If // if pixel is not in the oblique view image, the (NO\_CORRESP\_VAL, NO\_CORRESP\_VAL) remains in the collocation grid. End For // Loop on A, B, TDI and 1 km sub-bands //  $(i_{{\it L1b}}^{{\it Su}}$  ,  $j_{{\it L1b}}^{{\it Su}}$  ) was an L1b orphan pixel in SLSTR Su channel Else  $C_{b}^{m}(k, j) = (ORPHAN, ORPHAN)$ , b=33, 36 and 34, 35 if necessary, where ORPHAN is a temporary exception value End If End For // Loop on OLCI pixels End For // Loop on camera Now treat the "orphan" pixels in  $C_b^m(k, j)$ , m=2 to 5, b=33 to 36 For OLCI camera module m = 2 to 5 For all (k,j), k=0 to N\_LINE\_OLC-1, j=0 to N\_DET\_CAM-1 For b=33, 36 and 34,35 (depending on SLST\_SWIR\_SELECT parameter)

If  $C_b^m(k, j) = (ORPHAN, ORPHAN)$  Then fill  $C_b^m(k, j)$  with the nearest valid value in

the  $C_b^m(k, j)$  grid as shown in Figure 3-17. "Valid" means that the value is nor ORPHAN

nor NO\_CORRESP\_VAL. The nearest value search must be restrained to distance of  $\pm$  COLLOC\_NN\_RADIUS pixels in rows and columns.

<sup>&</sup>lt;sup>5</sup> 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) according to SLSTR ATBD [RD-2], whatever the convention adopted in SLSTR SY-4 [RD-3].



**ISSUE:** 11 Page: 89/107

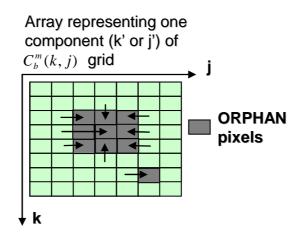
If the nearest neighbour search fails Then  $C_b^m(k, j) = (NO\_CORRESP\_VAL, j)$ 

\_ NO\_CORRESP\_VAL)

End For End For

End For

Note: For nearest neighbor search, when several candidates are possible the one on the same row k (and with smallest j) is preferred.



# Figure 3-17: Gaps due to orphan pixels in SLSTR nadir view Su channel are filled by duplicating the nearest neighbour valid pixel.

# 3.3.4.5 Selecting the part of the SLSTR nadir view image covered by OLCI image

The objective of this processing is to select the part of the SLSTR nadir view image acquired in the OLCI/SLSTR common swath, more precisely than was done in the pre-processing stage (section 3.1.4.3.2.1) and in particular across-track borders.

**Step 1**: Find the corners of the rectangle in the SLSTR nadir view  $S_{b'}$  channel (with **b' referring to a 1km channel**) encompassing the five OLCI camera module images:

Read the first (k = 0) and last (k = N\_LINE\_OLC - 1) lines of the 5  $C_{21+b'}^m(k, j)$  grids. This gives a list of (k<sup>21+b'</sup>, j<sup>21+b'</sup>) coordinates in the S<sub>b'</sub> channel.

Read the first column (j = 0) of the  $C_{21+b'}^1(k, j)$  grid and the last column (j = N\_DET\_CAM - 1) of the  $C_{21+b'}^5(k, j)$  grid. Concatenate the obtained lists of  $(k^{21+b'}, j^{21+b'})$  values to the previous list. Find the minimum and maximum  $k^{21+b'}$  (resp.  $j^{21+b'}$ ) values in the previous concatenated list:  $k_{\min}^{21+b'}$  and  $k_{\max}^{21+b'}$  (resp.  $j_{\min}^{21+b'}$  and  $j_{\max}^{21+b'}$ ) If  $k_{\min}^{21+b'}$  mod 2 = 1 then  $k_{\min}^{21+b'} = k_{\min}^{21+b'} - 1$ If  $k_{\max}^{21+b'}$  mod 2 = 0 then  $k_{\max}^{21+b'} = k_{\max}^{21+b'} + 1$ 

**Step 2**: Select the corresponding part of the SLSTR nadir view channels with margin, and corresponding annotations (including time-stamps):

• Select the part of the SLSTR nadir view 1km channels defined by:

$$j_{\min}^{21+b'} - SLST \_ 1km \_ J \_ MARGIN \le j \le j_{\max}^{21+b'} + SLST \_ 1km \_ J \_ MARGIN$$



**ISSUE:** 11 Page: 90/107

 $k_{\min}^{21+b'} - SLST \_ 1km \_ K \_ MARGIN \le k \le k_{\max}^{21+b'} + SLST \_ 1km \_ K \_ MARGIN$ 

and the corresponding nadir view per pixel (1km) annotations.

• Select the part of the SLSTR nadir view 500m channels defined by:

 $2.(j_{\min}^{21+b'} - SLST \_ 1km \_ J \_ MARGIN) \le j \le 2.(j_{\max}^{21+b'} + SLST \_ 1km \_ J \_ MARGIN) + 1$ 

 $2.(k_{\min}^{21+b'} - SLST \_ 1km \_ K \_ MARGIN) \le k \le 2.(k_{\max}^{21+b'} + SLST \_ 1km \_ K \_ MARGIN) + 1$ 

and the corresponding nadir view per pixel (500m) annotations.

These selected part of the SLSTR nadir view channels and annotations are stored in the L1c product.

The size of the SLSTR nadir view 500m channels stored in the L1c product is:

N\_SCAN\_SLST\_NAD\_05km\_L1C x N\_PIX\_SLST\_NAD\_05km\_L1C with

N\_SCAN\_SLST\_NAD\_05km\_L1C =  $2(k_{max}^{21+b'} - k_{min}^{21+b'}) + 4.SLST_1km_K_MARGIN + 2$  and

 $N_{\text{PIX}}SLST_{\text{NAD}}05km_{\text{L1C}} = 2.(j_{\text{max}}^{21+b'} - j_{\text{min}}^{21+b'}) + 4.SLST_{1km}J_{\text{MARGIN}} + 2$ 

The size of the SLSTR nadir view 1km channels stored in the L1c product is:

N\_SCAN\_SLST\_NAD\_1km\_L1C x N\_PIX\_SLST\_NAD\_1km\_L1C with

N\_SCAN\_SLST\_NAD\_1km\_L1C =  $k_{\text{max}}^{21+b'} - k_{\text{min}}^{21+b'} + 2.SLST_1km_K_MARGIN + 1$  and

 $N_PIX\_SLST\_NAD\_1km\_L1C = j_{max}^{21+b'} - j_{min}^{21+b'} + 2.SLST\_1km\_J\_MARGIN + 1$ 

The nadir view 500m or 1km subsampled annotations and variables in the additional datasets (see section 3.1.4.3.3.2.2) are also cut at this stage if necessary.

**Step 3**: Set offsets to retrieve the original instrument scan and relative pixel numbers: The offset allowing to retrieve the original instrument scan number from the line number in the extracted SLSTR nadir view 1km channels is L1C\_SCAN\_OFFSET\_NAD\_1km =  $s_{\min}^{n,1km} + k_{\min}^{21+b'}$  -

SLST\_1km\_K\_MARGIN.  $s_{min}^{n,1km}$  is defined in paragraph 3.1.4.3.2.1.

The offset allowing to retrieve the original relative pixel number from the column number in the extracted SLSTR nadir view 1km channels is L1C\_PIX\_OFFSET\_NAD\_1km =  $j_{min}^{21+b'}$  - SLST 1km J MARGIN.

The offset allowing to retrieve the original instrument scan number from the line number in the extracted SLSTR nadir view 500m channels is L1C\_SCAN\_OFFSET\_NAD\_05km =

 $s_{\min}^{n,05km}$  +2.( $k_{\min}^{21+b'}$  - SLST\_1km\_K\_MARGIN).  $s_{\min}^{n,05km}$  is defined in paragraph 3.1.4.3.2.1.

The offset allowing to retrieve the original relative pixel number from the column number in the extracted SLSTR nadir view 500m channels is L1C\_PIX\_OFFSET\_NAD\_05km = 2.( $j_{min}^{21+b'}$  - SLST\_1km\_J\_MARGIN).

These offsets are included in the L1c product.

**Step 4**: Adapt the correspondence grids  $C_d^m(k, j)$  to the selected part of SLSTR channels For each SLSTR nadir view 1km channel (d = 28 to 32) do  $C_d^m(k, j) = C_d^m(k, j) - (k_{\min}^{21+b'} - SLST_1km_K_MARGIN, j_{\min}^{21+b'} - SLST_1km_J_MARGIN)$ For each SLSTR nadir view 500m channel (d = 22 to 27) do  $C_d^m(k, j) = C_d^m(k, j) - 2.(k_{\min}^{21+b'} - SLST_1km_K_MARGIN, j_{\min}^{21+b'} - SLST_1km_J_MARGIN)$ These grids are included in the L1c product.



# 3.3.4.6 Selecting the part of the SLSTR oblique view image covered by OLCI image

The objective of this processing is to select the part of the SLSTR oblique view image acquired in the OLCI/SLSTR common swath, more tightly than was done in the previous stages (sections 3.1.4.3.2.1) and in particular across-track borders. Note that since the West edge of the SLSTR oblique view image is included in the OLCI swath, this edge of the image is left as it is.

**Step 0**: Retrieve the rotation sense of the SLSTR oblique view scan:

- a. Read the ortho-rectified longitude of pixels (N\_SCAN\_SLST\_ALT\_05km\_CUT/2, 0) and (N\_SCAN\_SLST\_ALT\_05km\_CUT/2, N\_PIX\_SCAN\_ALT\_05km 1), respectively noted  $\varphi_0$  and  $\varphi_N$ .
- b. If  $\phi_0 \cdot \phi_N > 0$  (the scan does not cross the -180/180 meridian) then
  - If  $\phi_{N} \phi_0 < 0$  then SCAN\_ROT\_DIR = "E -> W" else SCAN\_ROT\_DIR = "W > E"
  - Else (the scan does crosses the -180/180 meridian)

If  $\phi_N \, \cdot \, \phi_0 < 0$  then SCAN\_ROT\_DIR = "W -> E" else SCAN\_ROT\_DIR = "E -> W"

End If

**Step 1**: Find the Eastern column, the minimum and the maximum lines of the SLSTR oblique view 1km channels included in OLCI camera module 5 image:

If SCAN\_ROT\_DIR = "W -> E"

Read the last column (j = N\_DET\_CAM - 1) of the ( $k^{1km}$ ,  $j^{1km}$ ) =  $C_{36}^5(k, j)$  grid, and find the maximum on  $j^{1km}$ , noted  $j_{max}^{1km}$ .

Else (// If SCAN ROT DIR = "E -> W")

Read the last column (j = N\_DET\_CAM – 1) of the ( $k^{1km}$ ,  $j^{1km}$ ) =  $C_{36}^5(k, j)$  grid, and find the minimum on  $j^{1km}$ , noted  $j_{min}^{1km}$ .

# End if

Read the first (k = 0) line of the 5 ( $k^{1km}$ , $j^{1km}$ ) =  $C_{36}^m(k, j)$  grids. This gives a list of ( $k^{1km}$ ,  $j^{1km}$ ) coordinates in the SLSTR oblique view 1km channels. Find the minimum  $k^{1km}$  in this list, noted  $k_{\min}^{1km}$ .

Read the last (k = N\_LINE\_OLC - 1) line of the 5 (k<sup>1km</sup>, j<sup>1km</sup>) =  $C_{36}^m(k, j)$  grids. This gives a list of (k<sup>1km</sup>, j<sup>1km</sup>) coordinates in the SLSTR oblique view 1km channels. Find the maximum k<sup>1km</sup> in this list, noted  $k_{max}^{1km}$ .

If  $k_{\min}^{1km} \mod 2 = 1$  then  $k_{\min}^{1km} = k_{\min}^{1km}$  -1 If  $k_{\max}^{1km} \mod 2 = 0$  then  $k_{\max}^{1km} = k_{\max}^{1km}$  +1

**Step 2**: Select the corresponding part of the SLSTR oblique view channels with margin, and corresponding annotations (including time-stamps):

• Select the part of the SLSTR oblique view 1km channels defined by:

 $0 \le j \le j_{\max}^{1km} + SLST_{1km} J_{MARGIN}$  if SCAN\_ROT\_DIR = "W -> E"

or  $max(0, j_{min}^{1km} - SLST \_ 1km \_ J \_ MARGIN) \le j \le N\_PIX\_SCAN\_ALT\_1km - 1$  if SCAN\_ROT\_DIR = "E -> W"



**ISSUE:** 11 Page: 92/107

 $k_{\min}^{1km} - SLST \_ 1km \_ K \_ MARGIN \le k \le k_{\max}^{1km} + SLST \_ 1km \_ K \_ MARGIN$ 

and the corresponding oblique view per pixel (1km) annotations.

• Select the part of the SLSTR oblique view 500m channels defined by:

 $0 \le j \le 2.(j_{\text{max}}^{1km} + SLST_{1km} J_{MARGIN}) + 1$  if SCAN\_ROT\_DIR = "W -> E"

or  $2 \cdot \max(0, j_{\min}^{1km} - SLST_{1km} J_{MARGIN}) \le j \le N_{PIX}SCAN_{ALT}O5km - 1$  if SCAN\_ROT\_DIR = "E -> W"

 $2.(k_{\min}^{1km} - SLST \_ 1km \_ K \_ MARGIN) \le k \le 2.(k_{\max}^{1km} + SLST \_ 1km \_ K \_ MARGIN) + 1$ 

and the corresponding oblique view per pixel (500m) annotations.

These selected part of the SLSTR oblique view channels and annotations are stored in the L1c product.

The size of the SLSTR oblique view 500m channels stored in the L1c product is: N\_SCAN\_SLST\_ALT\_05km\_L1C x N\_PIX\_SLST\_ALT\_05km\_L1C with N\_SCAN\_SLST\_ALT\_05km\_L1C =  $2.(k_{max}^{1km} - k_{min}^{1km}) + 4.SLST_1km_K_MARGIN + 2$  and N\_PIX\_SLST\_ALT\_05km\_L1C =  $2.(j_{max}^{1km} + SLST_1km_J_MARGIN) + 2$  if SCAN\_ROT\_DIR = "W -> E" N\_PIX\_SLST\_ALT\_05km\_L1C =  $N_PIX_SCAN_ALT_05km - 2.(j_{min}^{1km} - SLST_1km_J_MARGIN)$ if SCAN\_ROT\_DIR = "E -> W"

The size of the SLSTR oblique view 1km channels stored in the L1c product is: N\_SCAN\_SLST\_ALT\_1km\_L1C x N\_PIX\_SLST\_ALT\_1km\_L1C with N\_SCAN\_SLST\_ALT\_1km\_L1C =  $k_{max}^{1km} - k_{min}^{1km} + 2.SLST_1km_K_MARGIN + 1$  and N\_PIX\_SLST\_ALT\_1km\_L1C =  $j_{max}^{1km} + SLST_1km_J_MARGIN + 1$  if SCAN\_ROT\_DIR = "W -> E" N\_PIX\_SLST\_ALT\_1km\_L1C =  $N_PIX_SCAN_ALT_1km - j_{min}^{1km} + SLST_1km_J_MARGIN$  if

SCAN\_ROT\_DIR = "E -> W"

The oblique view 500m or 1km subsampled annotations and variables in the additional datasets (see section 3.1.4.3.3.2.2) are also cut at this stage if necessary

**Step 3**: Set offsets to retrieve the original instrument scan and relative pixel numbers: The offset allowing to retrieve the original instrument scan number from the line number in the extracted SLSTR oblique view 1km channels is L1C\_SCAN\_OFFSET\_ALT\_1km =  $s_{\min}^{a,1km} + k_{\min}^{1km}$  - SLST\_1km\_K\_MARGIN.  $s_{\min}^{a,1km}$  is defined in paragraph 3.1.4.3.2.1. The offset allowing to retrieve the original relative pixel number from the column number in the

extracted SLSTR oblique view 1km channels is: • L1C PIX OFFSET ALT 1km = 0 if SCAN ROT DIR = "W -> F" since the oblique

- L1C\_PIX\_OFFSET\_ALT\_1km = 0, if SCAN\_ROT\_DIR = "W -> E", since the oblique view image is not cut on its West edge.
- L1C\_PIX\_OFFSET\_ALT\_1km = max(0, j<sup>1km</sup><sub>min</sub> SLST\_1km\_J\_MARGIN), if SCAN\_ROT\_DIR = "E -> W",



**ISSUE:** 11 Page: 93/107

The offset allowing to retrieve the original instrument scan number from the line number in the extracted SLSTR oblique view 500m channels is L1C\_SCAN\_OFFSET\_ALT\_05km =  $s_{\min}^{a,05km}$  +2.( $k_{\min}^{1km}$  - SLST\_1km\_K\_MARGIN).  $s_{\min}^{a,05km}$  is defined in paragraph 3.1.4.3.2.1.

The offset allowing to retrieve the original relative pixel number from the column number in the extracted SLSTR oblique view 500m channels is:

- L1C\_PIX\_OFFSET\_ALT\_05km = 0, if SCAN\_ROT\_DIR = "W -> E", since the oblique view image is not cut on its West edge.
- L1C\_PIX\_OFFSET\_ALT\_05km = 2. max(0, j<sup>1km</sup> SLST\_1km\_J\_MARGIN), if SCAN\_ROT\_DIR = "W -> E",

These offsets are included in the L1c product.

**Step 4**: Adapt the correspondence grids  $C_d^m(k, j)$  (d= 33, 36 and 34, 35 if necessary) to the selected part of SLSTR channels:

For SLSTR oblique view 1km channels (d = 36) do  $C_{36}^m(k, j) = C_{36}^m(k, j) - (k_{\min}^{1km} - k_{\min}^{1km})$ 

SLST\_1km\_K\_MARGIN , L1C\_PIX\_OFFSET\_ALT\_1km) if

 $C_{36}^{m}(k, j) \neq (\text{NO}_{\text{CORRESP}}, \text{VAL}, \text{NO}_{\text{CORRESP}}, \text{VAL})$ 

For each SLSTR oblique view 500m channels (d = 33 to 35) do  $C_d^m(k, j) = C_d^m(k, j) - 2.(k_{\min}^{1km} - SLST_1km_K_MARGIN, L1C_PIX_OFFSET_ALT_05km)$  if

 $C_d^m(k, j) \neq (\text{NO}_\text{CORRESP}_\text{VAL}, \text{NO}_\text{CORRESP}_\text{VAL})$ 

These grids are included in the L1c product.

# 3.4 Construction of L1c annotations

# 3.4.1 Algorithm Inputs

# 3.4.1.1 OLCI and SLSTR L1b Annotations

These are the annotations retrieved during the pre-processing stage (sections 3.1.4.2.3 and 3.1.4.3.3) and possibly cut (section 3.3). Annotations are divided in per pixel annotations attached to the images in their acquisition geometry, on the L1c grids defined in paragraph 2.1, and sub-sampled annotations, link with geolocation information corresponding to tie point of the L1b products.

# 3.4.1.2 L1c OLCI/SLSTR mis-registration and collocation grids

These are the correspondence grids obtained in section 3.3 between OLCI reference channel and all other OLCI and SLSTR channels.

# 3.4.1.3 L1c Tie- Points statistics

For each camera module:

- Initial Number of tie points in the OLCI image (retrieved during pre-processing): N\_TP\_L1C(m)
- Number of tie-points actually used for deformation model estimation: N\_TP\_L1C\_END(m)



**Issue:** 11 Page: 94/107

# 3.4.2 Processing Objective

The objectives of the annotation processing module is to collect OLCI and SLSTR L1b annotations, adding L1c specific annotations and write the annotations in the specific data sets as defined in [RD-5].

# 3.4.3 Algorithm Outputs

Level 1c product annotations, to be included in the level 1c product, as defined in [RD-5].

# 3.4.4 Mathematical Description

The list of L1c product annotations, with the way they are obtained and how they are gridded in the L1c product, are given in [RD-5].

# 3.4.4.1 L1c annotations from L1b OLCI annotations

The processing has already been done during the pre-processing stage (see section 3.1.4.2). It concerns:

- per pixel annotations (section 3.1.4.2.2):
  - Ortho-rectified geolocation: longitude, latitude and altitude corrected from DEM.
  - Quality flags according to the list provided in the L1c definition [RD-5].
- Time-stamps
- Sub-sampled annotations (section 3.1.4.2.3):
  - o Geolocation, Sun and Viewing Geometry
  - Meteo annotations;
- Additional Data Sets.
  - Instrument Data

# 3.4.4.2 L1c annotations from SLSTR annotations

The processing has already been done during the pre-processing stage (see section 3.1.4.3). It concerns:

- per pixel annotations (section 3.1.4.3.2):
  - Quality flags according to the list provided in the L1c definition [RD-5].
- Time-stamps
- Sub-sampled annotations (section 3.1.4.3.3):
  - Geolocation and Viewing Geometry annotations;
  - Additional Data Sets.
- Additional Data Sets
  - Thermal infrared quality ADS
  - Visible and shortwave infrared quality ADS



**ISSUE:** 11 Page: 95/107

3.4.4.3 L1c specific annotations

3.4.4.3.1 Misregistration annotations

All the grids  $(k^d, j^d) = C_d^m(k, j)$  computed by the correspondence grids computation module (section 3.3) are reproduced in the L1c product, on the OLCI Pixel Resolution (PR) grids.

3.4.4.3.2 L1c Tie points Quality Indicator

The ratio (in percent) between the initial and final (after possible rejection throughout the processing) number of tie points in the 5 OLCI images is computed and stored as a quality indicator in the L1c product.

R\_TP\_OK(m) =100 \* N\_TP\_L1C\_END(m) / N\_TP\_L1C(m)



**ISSUE:** 11 Page: 96/107

# **APPENDICES**



**Issue:** 11 Page: 97/107

# A. APPENDIX A - COMPUTATIONAL DEFINITION OF DIRECT AND INVERSE ORTHORECTIFIED GEOLOCATION FUNCTIONS

This annex describes in a general way how to compute the direct and inverse ortho-rectified geolocation functions from an ortho-rectified geolocation grid. These functions are needed at several stages of the L1c processing. They are computed from the ortho-rectified geolocation information included in the L1b products, namely the direct ortho-rectified geolocation grid that give the longitude, latitude (and altitude) corrected from relief for each spatial pixel of the products.

The direct ortho-rectified geolocation function is written  $(\lambda, \phi) = loc(k^*, j^*)$ , where  $(\lambda, \phi)$  are longitude, latitude coordinates and  $(k^*, j^*)$  are real coordinates in an image. The inverse orthorectified geolocation function is written  $(k^*, j^*) = loc^{-1}(\lambda, \phi)$ . These correspondences take into account the effect of the relief trough the use of a Digital Elevation Model (DEM) provided as CFI. The altitude of the terrain point can be retrieved using the DEM:  $h = DEM(\lambda, \phi)$ . The DEM is not required at level 1c since the L1b ortho-rectified geolocation grids already take it into account.

# A.1 Direct Ortho-rectified Geolocation Function

# A.1.1 Algorithm Inputs

# A.1.1.1 An ortho-rectified geolocation grid

This grid represents the values of the ortho-rectified geolocation function sampled on a regular grid of integer coordinates (k,j) with k = 0,..., K-1 and j = 0,...,J-1. It can be written formally  $(\lambda, \varphi, h)_{kj}$ . The lat/lon are given in degree unit.

# A.1.1.2 Location $(k^*, j^*)$ to be ortho-geolocated

(k\*,j\*) are (sub)pixel coordinates in an image.

# A.1.1.3 Processing Inputs

LOC\_FCT\_Params: Interpolation method and parameters for bi-linear or bi-cubic interpolator, according to user choice.

# A.1.2 Algorithm Outputs

The ortho-rectified geolocation of  $(k^*, j^*)$ :  $(\lambda_{k^*, j^*}, \phi_{k^*, j^*}, h_{k^*, j^*})$ .

# A.1.3 Mathematical Description

An approximation of the value of the direct ortho-rectified geolocation function at point (k\*,j\*) can be computed from the ortho-rectified geolocation grid using simple interpolation methods. A bicubic interpolation method is used to compute  $\lambda_{k^*,j^*}$ ,  $\phi_{k^*,j^*}$  and  $h_{k^*,j^*}$  from the neighboring samples



**Issue:** 11 Page: 98/107

in the grid  $(\lambda, \varphi, h)_{kj}$ . A special care is taken when the calculus involves pixels on both part of the  $-180/180^{\circ}$  longitude limit.

**Direct\_loc** function: ortho-rectified geolocation function from an ortho-rectified geolocation grid

#### Inputs:

 $\begin{array}{l} Loc\_Tab\_x, \ Loc\_Tab\_y, \ Loc\_Tab\_h: \ direct \ ortho-rectified \ geolocation \ grid: \ (\lambda_{kj}, \phi_{kj}, h_{kj}) \\ = \ (Loc\_Tab\_x(k,j), \ Loc\_Tab\_y(k,j), \ Loc\_Tab\_h(k,j)) \ (degree \ unit) \\ k\_in, \ j\_in: \ (sub)pixel \ position \ (k^*, \ j^*) \ in \ the \ location \ grid \ where \ the \ ortho-rectified \ geolocation \ is \ to \ be \ calculated \ (real \ numbers) \end{array}$ 

#### Outputs:

```
x_est, y_est, h_est: position (lat/lon) on Earth corresponding to (sub)pixel (k_in,j_in): loc(k_in,j_in)
```

```
Description:
```

 $lon180_flag = 0$ 

```
Extract a neighborhood of size 2xR_NEIGH+1 pixels centered at (kc,jc) = round
(k_in,j_in) in the ortho-rectified geolocation grid. The size shall be greater than
the interpolation window (see next statements):
     Loc_Tab_x_ct(r,l) = Loc_Tab_x(kc-R_NEIGH+r,jc-R_NEIGH+l),
     Loc_Tab_y_ct(r,l) = Loc_Tab_y(kc-R_NEIGH+r,jc-R_NEIGH+l),
     Loc_Tab_h_ct(r,l) = Loc_Tab_h(kc-R_NEIGH+r,jc-R_NEIGH+l), for r,l =
0...2xR NEIGH;
R NEIGH depends on the chosen interpolation method:
      R_NEIGH = 1 for bi-linear interpolation
      R NEIGH = 2 for bi-cubic interpolation
If Loc_Tab_x_ct contains both negative values close to -180 and positive values close
to 180 then
      add 360 to all negative values of Loc_Tab_x_ct
      lon180_flag = 1
end if
Find x_est by interpolation in the 2D table loc_Tab_x_ct at position (k_in-
kc+R_NEIGH,j_in-jc+R_NEIGH);
Find y_est by interpolation in the 2D table loc_Tab_y_ct at position (k_in-
kc+R_NEIGH,j_in-jc+R_NEIGH);
Find h_est by interpolation in the 2D table loc_Tab_h_ct at position (k_in-
kc+R_NEIGH,j_in-jc+R_NEIGH);
If lon180_flag = 1 then x_est = x_est - 360;
Return (x_est, y_est, h_est);
```



# A.2 Inverse Ortho-rectified Geolocation Function

# A.2.1 Algorithm Inputs

# A.2.1.1 An ortho-rectified geolocation grid

This grid represents the values of the ortho-rectified geolocation function sampled on a regular grid of integer coordinates (k,j) with k = 0,..., K-1 and j = 0,...,J-1. It can be written formally  $(\lambda,\phi)_{kj}$ .

# A.2.1.2 Location $(\lambda, \varphi)$ for which to compute the inverse ortho-rectified geolocation

 $\lambda, \phi$  are latitude, longitude coordinates, corrected from a Digital Elevation Model.

# A.2.1.3 A first guess of the solution $(k_0^*, j_0^*)$

 $k_0^*, j_0^*$  are (sub)pixel coordinates in the image of a first guess of the inverse ortho-rectified geolocation of  $(\lambda, \phi)$ :  $(k_0^*, j_0^*) \approx Loc^{-1}(\lambda, \phi)$ . This is necessary to initialize the algorithm. The better this first guess will be, the quicker the algorithm will converge.

#### A.2.1.4 Processing Parameters

- TOL\_UNIT\_SWITCH: switch indicating if the tolerance on precision is specified as a tolerance in lat/lon or in meters on ground
- INVLOC\_TOL\_LATLON or INVLOC\_TOL\_GRD\_DIST : required precision on the solution (in lat/lon or in meter, according to TOL\_UNIT\_SWITCH parameter)
- N\_ITER\_MAX: maximum number of iterations allowed to find a solution
- EPSILON\_INV\_LOC
- OUTSIDE\_GRID\_VAL
- INVLOC\_NO\_CVG\_VAL
- INVLOC\_JCB\_ERR VAL

# A.2.2 Algorithm Outputs

- The inverse ortho-rectified geolocation of  $(\lambda, \varphi)$ :  $(k^*_{\lambda, \varphi}, j^*_{\lambda, \varphi}) = Loc^{-1}(\lambda, \varphi)$ .
- conv\_flag: flag indicating success or failure of the function

The following data shall be output as intermediate verification/analysis data by the Processor (see algorithm in the following section):

- The x[iter] and y[iter] at each iteration in the function Inv\_Loc.
- The error max\_err at each iteration in the function Inv\_Loc.
- The number of iterations iter before convergence is reached
- detJ: the determinant of the Jacobian in the function Inv\_Loc

ThalesAlenia		REFERENCE:         S3-DD-TAF-SY-0062           Date:         17/02/2015	
A Thales / Finmeccanica Company Space	ISSUE:	11	Page: 100/107

# A.2.3 Mathematical Description

The computation of the inverse ortho-rectified geolocation  $(k^*,j^*)$  of a point  $(\lambda,\phi)$   $(h = DEM(\lambda,\phi))$  is an iterative process that makes use of the direct ortho-rectified geolocation function described above. The core of the algorithm is based on Newton-Raphson method to find the solution  $(k^*_{\lambda,\phi}, j^*_{\lambda,\phi})$  of the non-linear equation Loc $(k^*, j^*) - (\lambda,\phi) = 0$ . The algorithm needs a first guess solution to start. The way this first guess is found should be described where the function is called.

A general trick can be used to find first guess solutions when computing inverse geolocation for a collection of targets coming from the same grid: After the solution of the first target point has been found, use this solution as a first guess for the neighboring targets, and so on from one target to the neighboring ones. The first guess for the first target point can be found by searching the nearest neighbor of  $(\lambda, \phi)$  among the  $(\lambda, \phi)_{kj}$  and taking the corresponding k,j as first guess.

The desired precision can be specified by the user as a tolerance on lat/lon coordinates or in meters on ground.

# Computation of the inverse ortho-rectified geolocation

The calculus is made by the following iterative algorithm:

Inv\_loc function: inverse ortho-rectified geolocation function

#### Inputs:

```
Loc\_Tab\_x, Loc\_Tab\_y: direct ortho-rectified geolocation grid (\lambda_{kj}, \varphi_{kj}) = (Loc\_Tab\_x(k,j), Loc\_Tab\_y(k,j)) (degree unit)
x\_target, y\_target, h\_target: position in the lat/lon coordinates whose antecedent is searched, with its altitude
k\_init, j\_init: initial guess
```

Outputs:

k\_est, j\_est: estimated (non integer) solution. Filled with OUTSIDE\_GRID\_VAL value if the algorithm has failed conv\_flag: flag indicating success or failure of the function. Possible values are: OUTSIDE\_GRID\_FLAG: the searched k\_est, j\_est are not in the input orthorectified geolocation grid INVLOC\_CVG\_OK\_FLAG: k\_est, j\_est have been found with the required precision INVLOC\_NO\_CVG\_FLAG: the computation of k\_est,j\_est did not reach the required precision in the N\_ITER\_MAX iterations INVLOC\_JCB\_ERR\_FLAG: the Jacobian matrix is ill-conditioned for inversion



**ISSUE:** 11 Page: 101/107

#### Description:

```
If Loc_Tab_x contains both negative values close to -180 and positive values close to
180 then add 360 to all negative values of Loc_Tab_x_ct
If x_target is negative and close -180 then add 360 to x_target
k[0] = k_{init};
j[0] = j_init;
eps = EPSILON_INV_LOC;
last_try = 0;
iter = 0;
while iter < N_ITER_MAX
      // Check if the search for the sub-pixel location goes outside the ortho-
      // rectified geolocation grid
      if (k[iter] < 0) or (k[iter] > K-1) or (j[iter] < 0) or (j[iter] > J-1) then
            k_est = OUTSIDE_GRID_VAL;
            j_est = OUTSIDE_GRID_VAL;
            conv_flag = OUTSIDE_GRID_FLAG;
            return (k_est,j_est,conv_flag);
      end if
      // Compute value of (x[iter],y[iter]) = Loc(k[iter],j[iter]) and corresponding
      // Jacobian:
      Jacobian(Loc_Tab_x, Loc_Tab_y, k[iter], j[iter], eps) ->
      (J11,J12,J22,J21,x[iter],y[iter]);
      detJ = J11*J22 - J12*J21; // determinant of the Jacobian
      if abs(detJ) < DET_J_COND // Test if Jacobian matrix is well conditioned
            if last try \neq 1
                  Jacobian(Loc_Tab_x, Loc_Tab_y, k[iter], j[iter], eps/10) ->
                  (J11,J12,J22,J21,x[iter],y[iter]);
                  last_try = 1;
            else
                  conv_flag = INVLOC_JCB_ERR_FLAG;
                  k_est = INVLOC_JCB_ERR VAL;
                  j_est = INVLOC_JCB_ERR VAL;
                  return (k_est,j_est,conv_flag);
            end if
      end if
      if TOL_UNIT_SWITCH == "LON/LAT" then
            max_err = max(abs(x[iter]-x_target),abs(y[iter]-y_target));
            INVLOC_TOL = INVLOC_TOL_LATLON
      end if
      if TOL_UNIT_SWITCH == "GRD_DIST" then
            max_err = Geodetic_distance((x[iter], y[iter]),(x_target, y_target),
            h_target)
            INVLOC_TOL = INVLOC_TOL_GRD_DIST
      end if
      if max_err < INVLOC_TOL then // a solution has been found with sufficient
                                   // precision
            k_est = k[iter];
            j est = j[iter];
            conv_flag = INVLOC_CVG_OK_FLAG;
            return (k_est,j_est,conv_flag);
```



**Issue:** 11 Page: 102/107

end if

```
// Newton-Raphson: [delta_k,delta_j]<sup>T</sup> = J<sup>-1</sup> x [delta_x, delta_y]<sup>T</sup>
delta_k = -(J22*(x[iter]-x_target) - J12*(y[iter]-y_target))/detJ;
delta_j = -(J11*(y[iter]-y_target) - J21*(x[iter]-x_target))/detJ;
k[iter+1] = k[iter] + delta_k;
j[iter+1] = j[iter] + delta_j;
iter = iter + 1;
End while
conv_flag = INVLOC_NO_CVG_FLAG;
k_est = INVLOC_NO_CVG_VAL;
j_est = INVLOC_NO_CVG_VAL;
return (k_est,j_est,conv_flag);
```

The function Geodetic\_distance(( $\lambda_1, \varphi_1$ ),( $\lambda_2, \varphi_2$ ), h) compute the geodetic distance, *at altitude h*, between the points ( $\lambda_1, \varphi_1$ ),( $\lambda_2, \varphi_2$ ) located on Earth in lat/lon coordinates.

The geodetic distance at altitude h is defined as the geodetic distance on the reference ellipsoid augmented with altitude h (major & minor axes a and b become a+h and b+h), being admitted that the term "geodetic distance" is not restricted to define a distance on the reference ellipsoid. Distance at altitude h is thus obtained with the same formula (not described here) as the one used to compute distance on the reference ellipsoid, but substituting a+h and b+h to a and b parameters. For Vincenty's formula, refer to <a href="http://www.movable-type.co.uk/scripts/latlong-vincenty.html">http://www.movable-type.co.uk/scripts/latlong-vincenty.html</a> and <a hre

Description of the function Jacobian:

```
Jacobian function: Compute the Jacobian matrix of the direct location function at a location (k_{in}, j_{in})
```

#### Inputs:

Loc\_Tab\_x, Loc\_Tab\_y: direct ortho-rectified geolocation grid
k\_in, j\_in: location where the Jacobian is to be calculated
eps: step used to compute the derivative of the location function

#### Outputs:

J11, J12, J22, J21: elements of the Jacobian matrix  $x\_loc$ ,  $y\_loc$ : values of the direct location function at (k\_in, j\_in)



**ISSUE:** 11 Page: 103/107

#### Description:

```
// compute the ortho-rectified geolocation function at (k_{in,j_in})
Direct_loc(Loc_Tab_x, Loc_Tab_y, k_in, j_in) -> (x_loc,y_loc) // see paragraph A.1
// Compute the partial derivative of the direct location function with respect to
// variable k:
k_d = k_i;
h = eps;
k_d = k_d + h;
Direct_loc(Loc_Tab_x, Loc_Tab_y, k_d, j_in) -> (x_loc_dk,y_loc_dk)
J11 = (x_loc_dk - x_loc)/h;
J21 = (y_loc_dk - y_loc)/h;
// Compute the partial derivative of the direct location function with respect to
// variable j:
j_d = j_{in};
h = eps;
j_d = j_d + h;
Direct_loc(Loc_Tab_x, Loc_Tab_y, k_in, j_d) -> (x_loc_dj,y_loc_dj)
J12 = (x_loc_dj - x_loc)/h;
J22 = (y_loc_dj - y_loc)/h;
Return (J11, J12, J22, J21, x_loc, y_loc);
```

A particular care must be taken when the longitude  $\lambda$  is close to 180° or -180°. Then 360° must be added to the negative  $\lambda_{kj}$  of the input grid (and 360° must be added to  $\lambda$  if it is negative) before to perform the iterative algorithm.

# **B. APPENDIX B - BI-CUBIC INTERPOLATION**

# **B.1** Algorithm inputs

- Let S(k,j) be a 2D function S(y,x) (e.g.: an image) sampled on a regular grid {(k,j), k = 1,2,...,K and j = 1,2,...,J}.
- A non-integer location (y,x) in the grid (k,j)
- Processing parameters:
  - o Interpolation parameter: a

# **B.2** Processing Objective

The objective of interpolation is to estimate the value S(y,x) at any non-integer location (y,x), from the S(k,j) values.

ThalesAlenia	Referenci Date:	REFERENCE:         S3-DD-TAF-SY-00620           DATE:         17/02/2015		
A Thales / Finmeccanica Company Space	ISSUE:	11	Page: 104/107	

#### **B.3** Mathematical description

The Shannon sampling theorem states that, under certain conditions the function S(y,x) can be expressed as a weighted sum of cardinal sine functions, the weights being the values of S at the sampled location (k,j). The bi-cubic interpolator (or cubic spline interpolators) approximates the cardinal-sine function by a set of splines. Instead of determining one high order polynomial passing through all the points, a set of low order polynomials (cubic) are used to approximate the cardinal-sine function over different intervals; the coefficient are chosen so that the function and its low order derivates match where the intervals meet. Bi-cubic spline image interpolation can be efficiently obtained by applying a convolution with the following kernel in both dimensions (from [K81]):

 $\begin{cases} Cubic4_a(t) = 1 - (a+3) t^2 + (a+2) t^3 & \text{if } 0 \leq t \leq 1 \\ Cubic4_a(t) = -4 a + 8 at - 5 a t^2 + a t^3 & \text{if } 1 \leq t \leq 2 \\ Cubic4_a(t) = 0 & \text{if } t > 2 \\ Cubic4_a(-t) = Cubic4_a(t) & \text{if } t > 2 \end{cases}$ 

where a is the slope at t=1. The value of parameter a must be tunable.

Let us write  $j_0 = FLOOR[x]$  and  $k_0 = FLOOR[y]$ , and  $dx = x-j_0$ ,  $dy = y-k_0$  ( $0 \le dx, dy < 1$ ). Then, consider the 4x4 neighbourhood of (x,y) in the (k,j) grid defined by {( $k_0+u, j_0+v$ ), with u,v = -1,0,1,2} as shown in Figure B-1.

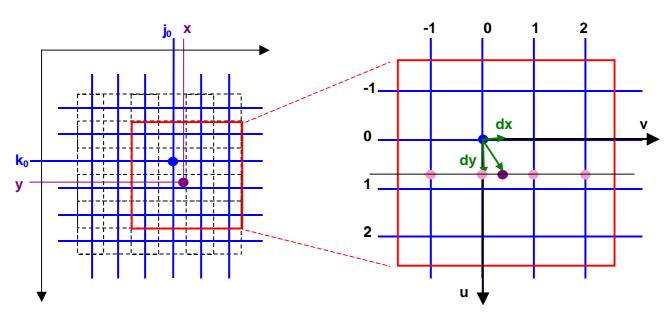


Figure B-1: Bi-cubic interpolation scheme. Left figure: intersection of blue lines are the location of the samples S(k,j). Dashed black squares represent the corresponding pixel area (if S(k,j) is an image). The purple point is the location of the searched value S(y,x). The red square is the 4x4 (pixel) neighbourhood of (x,y). The right figure concentrates on the 4x4 neighbourhood re-indexed with (u,v) local coordinates. The searched value S(x,y) is obtained by first 1D-interpolating the columns at location dy (this gives the 4 pink points) and then 1D-interpolating the obtained sampled row at location dx.



The interpolation coefficients are then:  $c_k(dy) = \text{Cubic4}_a(k-dy)$  and  $c_j(dx) = \text{Cubic4}_a(j-dx)$ .

Thus, the value of S(y,x) is obtained by computing a weighted linear combination of the nearest neighbours of (y,x):

$$S(y,x) = S(k_0 + dy, j_0 + dx) = \sum_{u=-1}^{2} \sum_{\nu=-1}^{2} c_u(dy) c_{\nu}(dx) S(k_0 + u, j_0 + \nu)$$

This can be computed in two steps of 1D interpolation: Compute, for v = -1,0,1,2:

$$S(y, j_0 + v) = S(k_0 + dy, j_0 + v) = \sum_{u=-1}^{2} c_u(dy) S(k_0 + u, j_0 + v)$$

This gives 4 interpolated values represented in pink in Figure B-1. Then compute the searched value,

$$S(y,x) = S(k_0 + dy, j_0 + dx) = \sum_{\nu=-1}^{2} c_{\nu}(dx) S(y, j_0 + \nu)$$

in purple in Figure B-1.

# **B.4** Algorithm Outputs

The interpolated value S(y,x).

# C. APPENDIX C - SHANNON TRUNCATED APODIZED INTERPOLATION

# C.1 Algorithm inputs

- Let S(k,j) be a 2D function S(y,x) (e.g.: an image) sampled on a regular grid {(k,j), k = 1,2,...,K and j = 1,2,...,J}.
- A non-integer location (y,x) in the grid (k,j)
- Processing parameters:
  - $\circ$  Interpolation parameters: values of  $\alpha$  and N

# C.2 Processing Objective

The objective of interpolation is to estimate the value S(y,x) at any non-integer location (y,x), from the S(k,j) values.

# C.3 Mathematical description

The Shannon sampling theorem states that, under certain conditions the function S(y,x) can be expressed as a weighted sum of cardinal sine functions, the weights being the values of S at the sampled location (k,j). (k,j). In practice the sum must be limited to a small number of terms. This is achieved limiting the support of the cardinal sine kernel to a size (2N+1)x(2N+1). To limit the artifact (Gibbs oscillations) induced by this truncation an apodization function is applied to the kernel.

The 1D interpolation kernel is:



 $h_{1D}(x) = \operatorname{sinc}(x)W(x)$ , for all x in [-N,N],

with 
$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$
 (sinc(0)=1)

and W is a Hann or Hamming window:

$$W(x) = \alpha + (1 - \alpha) \cos(\pi x / N)$$

where  $\alpha = 0.5$  for a Hann apodization window and  $\alpha = 0.54$  for Hamming apodization window.

Let us write  $j_0 = FLOOR[x]$  and  $k_0 = FLOOR[y]$ , and  $dx = x-j_0$ ,  $dy = y-k_0$  ( $0 \le dx, dy < 1$ ).

The 2D interpolation formula is:

$$S(y,x) = S(k_0 + dy, j_0 + dx) = \sum_{u=-N}^{N} \sum_{v=-N}^{N} h_{1D}(dy - u) h_{1D}(dx - v) S(k_0 + u, j_0 + v)$$

This can be computed in two steps of 1D interpolation: Compute, for v = -N to N:

$$S(y, j_0 + v) = S(k_0 + dy, j_0 + v) = \sum_{u=-N}^{N} h_{1D} (dx - u) S(k_0 + u, j_0 + v)$$

This gives 2N+1 interpolated values  $S(y,j_0+v)$ . Then compute the searched value,

$$S(y,x) = S(k_0 + dy, j_0 + dx) = \sum_{v=-N}^{N} h_{1D} (dy - v) S(y, j_0 + v)$$

# C.4 Algorithm Outputs

The interpolated value S(y,x).



 REFERENCE:
 S3-DD-TAF-SY-00620

 DATE:
 17/02/2015

**ISSUE:** 11 Page: 107/107

END OF DOCUMENT