SMOS L1PP

# Analytical Pixel Footprint

| | | |
|---|---|---|
| **Code** | : | SO-TN-DME-L1PP-0172 |
| **Issue** | : | 1.0 |
| **Date** | : | 18/04/08 |

| | **Name** | **Function** | **Signature** |
|---|---|---|---|
| **Prepared by** | Rita Castro | Trainee | |
| **Checked by** | Nuno Catarino | Project Engineer | |
| **Approved by** | José Barbosa | Project Manager | |

DEIMOS Engenharia
Av. D. João II, Lote 1.17.01 Torre Zen, 10º,
1998-023 Lisboa, PORTUGAL
Tel: +351 21 893 30 10
Fax: +351 21 896 90 99
E-mail: mailto:deimos@deimos.com.pt

This page intentionally left blank

# Document Information

| Contract Data | Classification |
|---|---|
| **Contract Number:** DE04/B-434/P | **Internal** ☑<br>**Public** ☐ |
| **Contract Issuer:** EADS CASA Espacio | **Industry** ☐<br>**Confidential** ☐ |

| Internal Distribution | | |
|---|---|---|
| Name | Unit | Copies |
| | | |

| External Distribution | | |
|---|---|---|
| Name | Organisation | Copies |
| | | |

| Archiving | |
|---|---|
| Word Processor: | MS Word 2007 |
| File Name: | SO-TN-DME-L1PP-0001-Footprint.doc |
| Archive Code: | P/TN/DME/03/013-029 |

# Document Status Log

| Issue | Change description | Date | Approved |
|---|---|---|---|
| 1.0 | Description and validation of the algorithm used in L1c's pixel footprint calculation | April 2008 | |

# 1. INTRODUCTION

## 1.1. Purpose and Scope

This document describes the processing approach to reduce the computation time on the footprint calculation on the Geolocation module.

The pixel footprint calculation is, essentially, a geometrical problem about changing and rotating reference frames and a similar algorithm approach can be developed to solve the Galaxy Map rotation problem.

## 1.2. Reference Documents

| Ref. | Code | Title | Issue |
|---|---|---|---|
| RD.01 | SO-LI-CASA-PLM-0094 | Directory of Acronyms and abbreviations | |
| RD.02 | SO-DS-DME-L1PP-0009 | SMOS L1Processor L1c Data Processing Model | *2.0* |
| RD.03 | SO-TR-DME-L1PP-0018 | SMOS L1 Prototype Software Verification and Validation Plan | *2.3* |
| RD.04 | CS-MA-DMS-GS-0001 | EE - Mission CFI Software<br>MISSION CONVENTIONS DOCUMENT | *1.3* |
| RD.05 | EE-MA-DMS-GS-0002 | EE - Mission CFI Software<br>GENERAL SUM | *3.7* |
| RD.06 | EE-MA-DMS-GS-0003 | EE - Mission CFI Software<br>EXPLORER_LIB SUM | *3.7* |
| RD.07 | EE-MA-DMS-GS-0004 | EE - Mission CFI Software<br>EXPLORER_ORBIT SUM | *3.7* |
| RD.08 | EE-MA-DMS-GS-0005 | EE - Mission CFI Software<br>EXPLORER_POINTING SUM | *3.7* |
| RD.09 | http://en.wikipedia.org/wiki/Matrix_representation_of_conic_sections | | |

*Table 1: Reference Documents*

# 2. ANALYTICAL PIXEL FOOTPRINT

## 1.3. Definitions

**Rotation Matrices:** Using the roll-pitch-yaw convention, the following rotation matrices apply:

$$R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_2(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \qquad R_3(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

***Eq. 1***

## 1.4. The geometrical problem

Calculating the pixel footprint can be described as the geometrical problem of finding the intersection of an elliptical cone, defined in a given coordinate system, with a plane defined in other coordinate system.

The cone is defined on a satellite frame whereas the plane is defined on a Earth-topocentric frame.

Let's start by defining the cone. Given a looking direction on the satellite, defined by an azimuth ($\varphi$) and an elevation *($\theta$),* (i.e., the S/C to pixel directions expressed on its frame) and

1) a circumference on the $xOy$ satellite plane[1] ;
2) unit semi-sphere of the satellite,

then the projection of the circumference of radius $r_c$ on the tangent plane to the unit sphere at the looking direction creates an elliptical cone (cf. Figure 1).

[1] This circumference corresponds to the -3dB contour of the Synthetic Antenna Directional Gain and will be referred to as $r_c$.

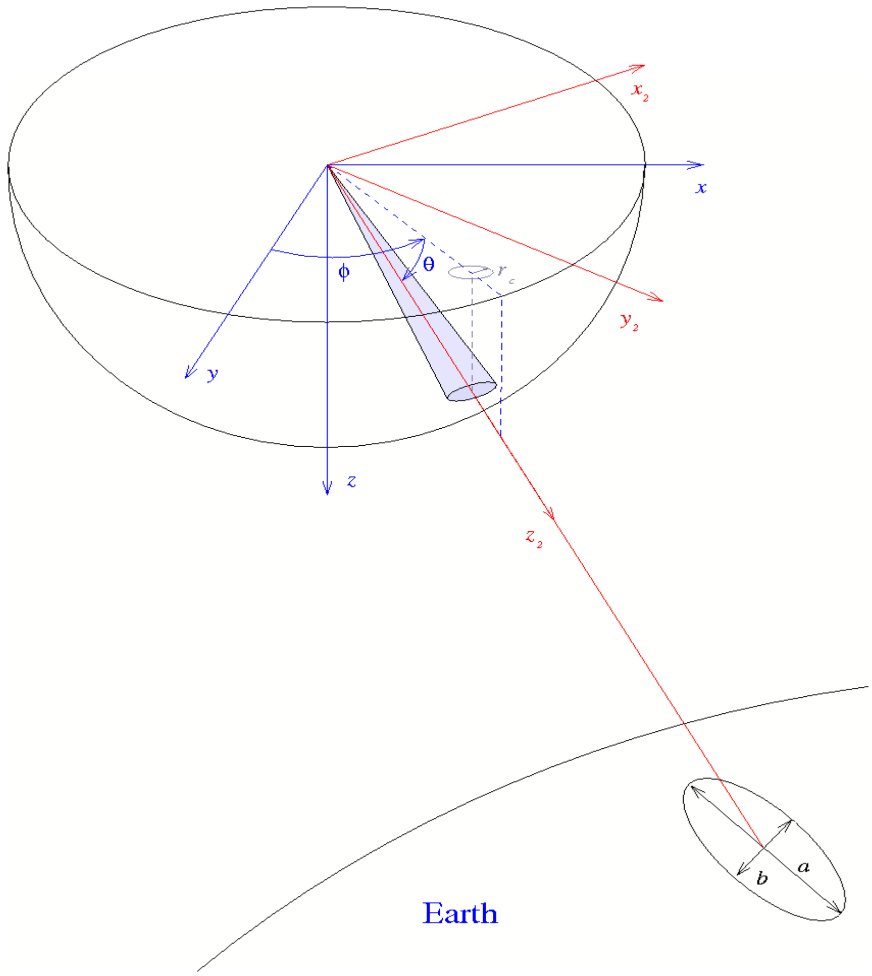



***Figure 1: Cone defined on the satellite frame and projection on to the Earth. In blue is the original reference frame on the satellite and in red it is the cone-aligned one. Other quantities are defined in the text.***

In order to write a simple Cartesian equation to describe the cone in the satellite frame two rotations can be performed:

1) Rotating the *x0y* plane in order to align the axis' cone with the $z_1 0 y_1$ plane.
   This results in a set of coordinates $(x_1, y_1, z_1)$ as in where $R_3(-\varphi)$ is the rotation matrix as in Eq. 1.
2) Rotating around $x_1$ axis with an angle of $(\theta - 90^o)$, creating a set of coordinates $(x_2, y_2, z_2)$ defined by

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = R_1(\theta - 90^\circ) R_3(-\varphi) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \textbf{Eq. 2}$$

In the *($x_2$, $y_2$, $z_2$)* referential, the oblique cone is aligned with the $z_2$ axis and its apex is on the origin. Therefore the equation of the cone is given by Eq. 3.

$$x_2^2 + \cos^2(\theta)\, y_2^2 - r_{-3dB}^2 z_2^2 = 0$$ ***Eq. 3 (a)***

$$X_2^T m X_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos^2(\theta) & 0 \\ 0 & 0 & -r_c^2 \end{bmatrix}$$ ***(b)***

The topocentric plane is defined in a local tangent plane on the Earth and is centred on the intersection of the looking direction, from now on mentioned as the line of sight (LOS); The $x$ axis points towards East, the $y$ axis points North and the $z$ axis points to the zenith, as shown in Figure 2.

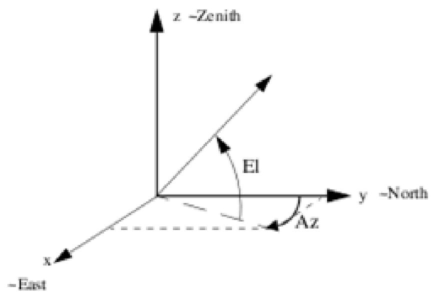



***Figure 2: Topocentric frame - axes and angles definitions***

A Cartesian vector in satellite frame, $X_{sat}$ can be represented as a rotation between the topocentric plane and the satellite frame plus the translation vector between the origins of the two CS's (Eq. 4)[2].

$$X_{sat} = M_{sat}^{topocentric} X_{topocentric} + T_{sat}^{topocentric}$$ ***Eq. 4***

Combining Eq. 2, Eq. 3 and Eq. 4 and setting $z = 0$ in the topocentric vector the intersection between the elliptic cone and the Earth tangent plane at the LOS is given by a conic equation (in topocentric coordinates).

The analysis of this conic section is done using matrices. Given a general conic,

$$Ax^2 + By^2 + Cx + Dy + Exy + F = 0$$ ***Eq. 5***

the matrix associated with this conic is

[2] On this TN any transformation between two frames is represented as follows: $K_{finalCS}^{initialCS}$, where $K$ can be a rotation matrix or a translation vector.

$$A_q = \begin{bmatrix} F & C/2 & D/2 \\ C/2 & A & E/2 \\ D/2 & E/2 & B \end{bmatrix} \qquad \text{Eq. 6}$$

For non degenerate conics the determinant of the sub-matrix

$$A_{11} = \begin{bmatrix} A & E/2 \\ E/2 & B \end{bmatrix} \qquad \text{Eq. 7}$$

determines what type of conic it is.

The eigenvalues of $A_{11}$ are used to write a reduced equation for the conic

$$\lambda_1 x'^2 + \lambda_2 y'^2 + \frac{|A_q|}{|A_{11}|} = 0 \qquad \text{Eq. 8}$$

And for an ellipse,

$$a = \sqrt{-\frac{|A_q|}{\lambda_1 |A_{11}|}}$$

$$b = \sqrt{-\frac{|A_q|}{\lambda_2 |A_{11}|}} \qquad \text{Eq. 9}$$

# 3. IMPLEMENTATION

In Section 1.4 the geometrical problem was presented without specifying the satellite referential in question. As described in [RD.02] the Best Fit Plane (BFP) is the referential where the antenna plane is located and from where the *(ξ, η)* are defined, therefore it makes sense to use it as the reference frame where the elliptic cone is defined. When the satellite attitude is initialized, a rotation matrix between an Earth frame and the BFP is defined – this matrix shall be one of the outputs of the attitude initialization functions in L1PP.

The transformations between EF and topocentric reference frames are performed by EE-CFI function `xl_ef_to_topocentric`. The rotation matrix between the satellite frame and the topocentric frame, $M_{sat}^{topocentric}$ in Eq. 4 is obtained by transforming the rotation matrix between the BFP and the EF, $M_{EF}^{BFP}$ by calling the EE-CFI function `xl_ef_to_topocentric`, with:

- **mode:** set to XL_MODE_FLAG_DIRECTION,
- **derive:** set to XL_NO_DER,
- **pos:** position of the LOS in EF coordinates,
- **ef_dir:** column vector of $M_{EF}^{BFP}$ matrix,
- **vel, ef_dir_d:** as dummy variables,

the topocentric angular coordinates, azimuth and elevation for each column vector is calculated. Converting them to Cartesian coordinates,

$$M_{BFP}^{topocentric} = \begin{bmatrix} \cos(\theta_i)\sin(\varphi_i) \\ \cos(\theta_i)\cos(\varphi_i) \\ \sin(\theta_i) \end{bmatrix}_{i=column1,column2,column3} \qquad \textbf{Eq. 10}$$

The translation between the BFP and the topocentric frame is calculated by applying `xl_ef_to_topocentric` with the same arguments as above, except:

- **mode:** set to XL_MODE_FLAG_LOCATION[3],
- **ef_dir:** satellite position in EF coordinates.

The azimuth, elevation and range are then used to calculate the Cartesian topocentric coordinates for the translation vector, $T_{sat}^{topocentric}$ in Eq. 4 as

$$T_{BFP}^{topocentric} = \begin{cases} \cos(\theta)\sin(\varphi) \\ \cos(\theta)\cos(\varphi) \\ \sin(\theta) \end{cases} \qquad \textbf{Eq. 11}$$

[3] If **mode** is set to XL_MODE_FLAG_LOCATION, then range is the distance between the origin of the topocentric CS and the point defined by ef_dir. However, if **mode** is set to XL_MODE_FLAG_DIRECTION, then range is unitary.

Using the top input variable **angles**, Eq. 2 and Eq. 3 b) are evaluated and with $M_{BFP}^{topocentric}$, and $T_{BFP}^{topocentric}$ a conic equation as in Eq. 5 is calculated. That equation is an ellipse whose semi-axes are extracted with Eq. 9.

# 4. RESULTS AND CONCLUSIONS

In order to validate the new algorithm the test scenario 1461 listed in [RD.03] was used. There are two products on this scenario: one with 700 scenes, Figure 3, the other with 2300 scenes, Figure 4.

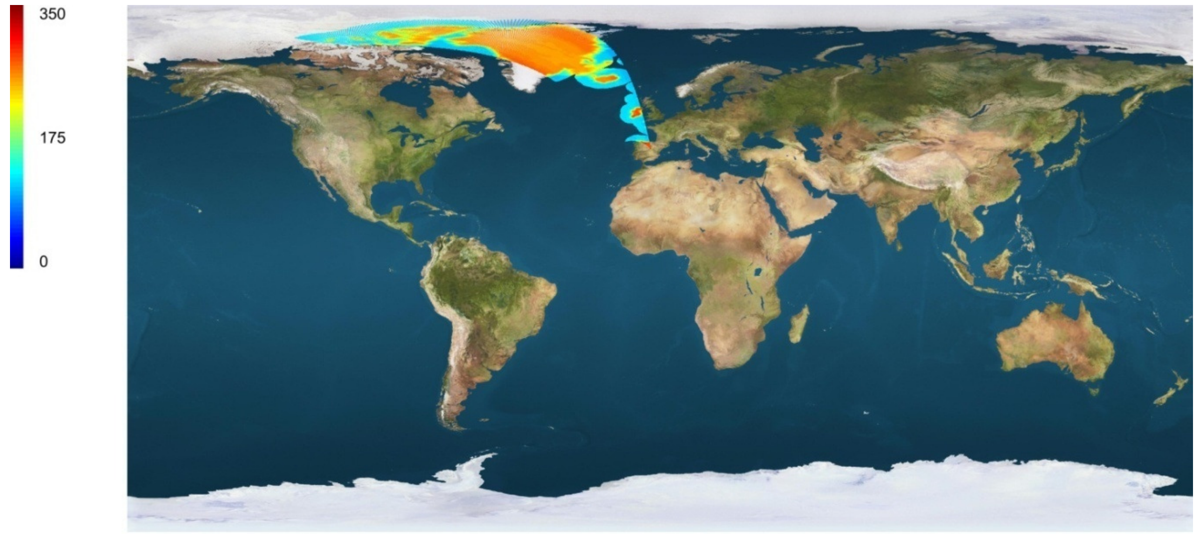



*Figure 3: First product of test 1461 - contains ~300 scenes.*

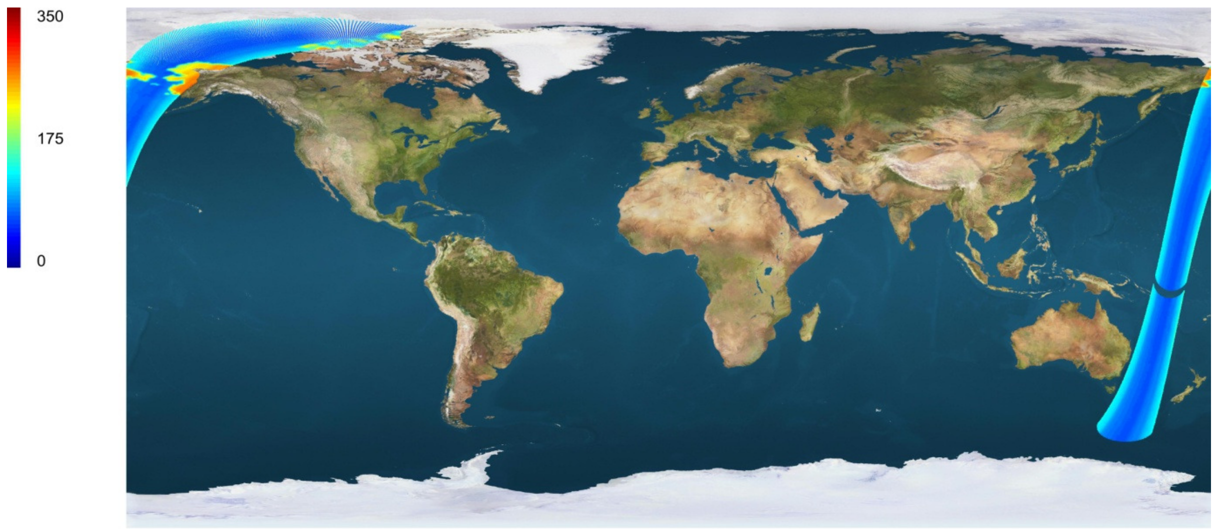
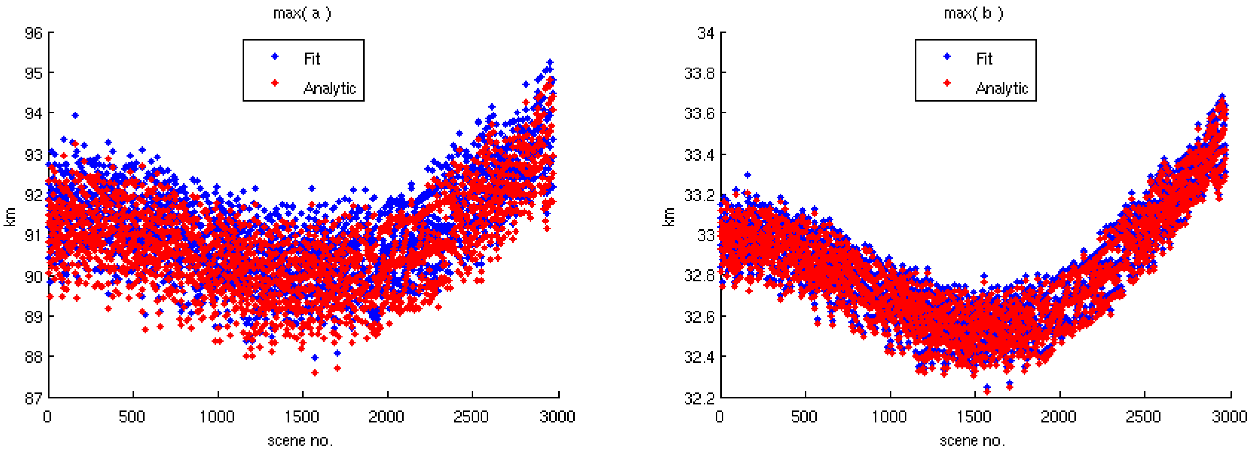



*Figure 4: First product of test 1461 - contains ~2700 scenes.*

It has been shown that:

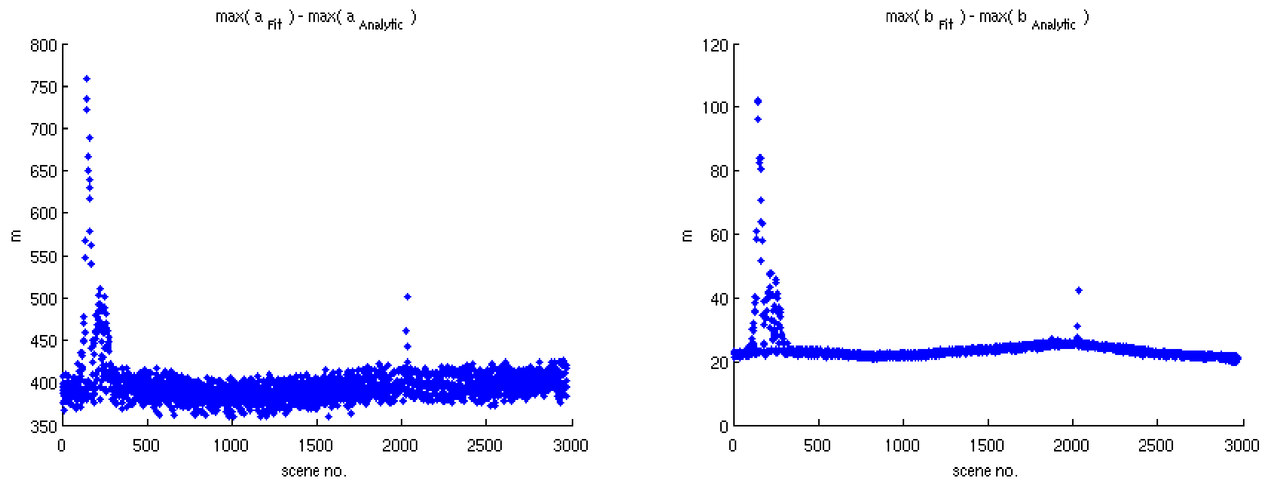
a) Processing time has reduced by a maximum factor of 50% (cf. Figure 8);

b) Nominal values obtained with an analytical calculation of *a* and *b* are essentially the same (cf. Figure 5) except in some scenes where the differences can span from 400 to 750 meters (in the major semi axe), or 20 to 100 meters (minor semi axe), roughly (cf. Figure 6 and Figure 7).




(a) (b)

***Figure 5: Comparison between maximum values for a and b.***




(a) (b)

***Figure 6: Absolute difference between maximum values of ellipse semi-axes calculated with both algorithms.***

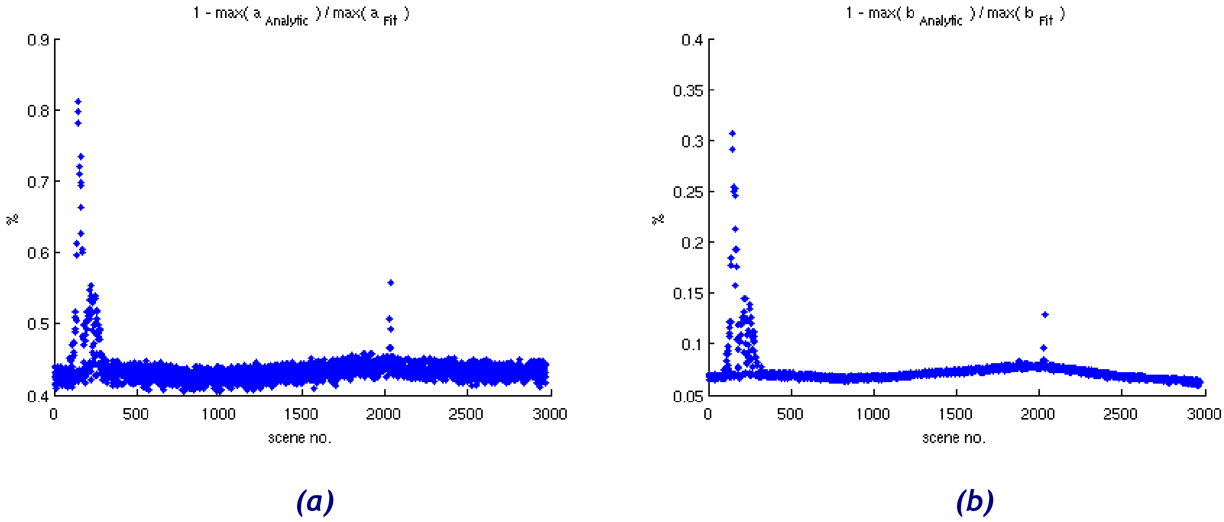



*Figure 7: Normalized difference between maximum values of ellipse semi-axes.*

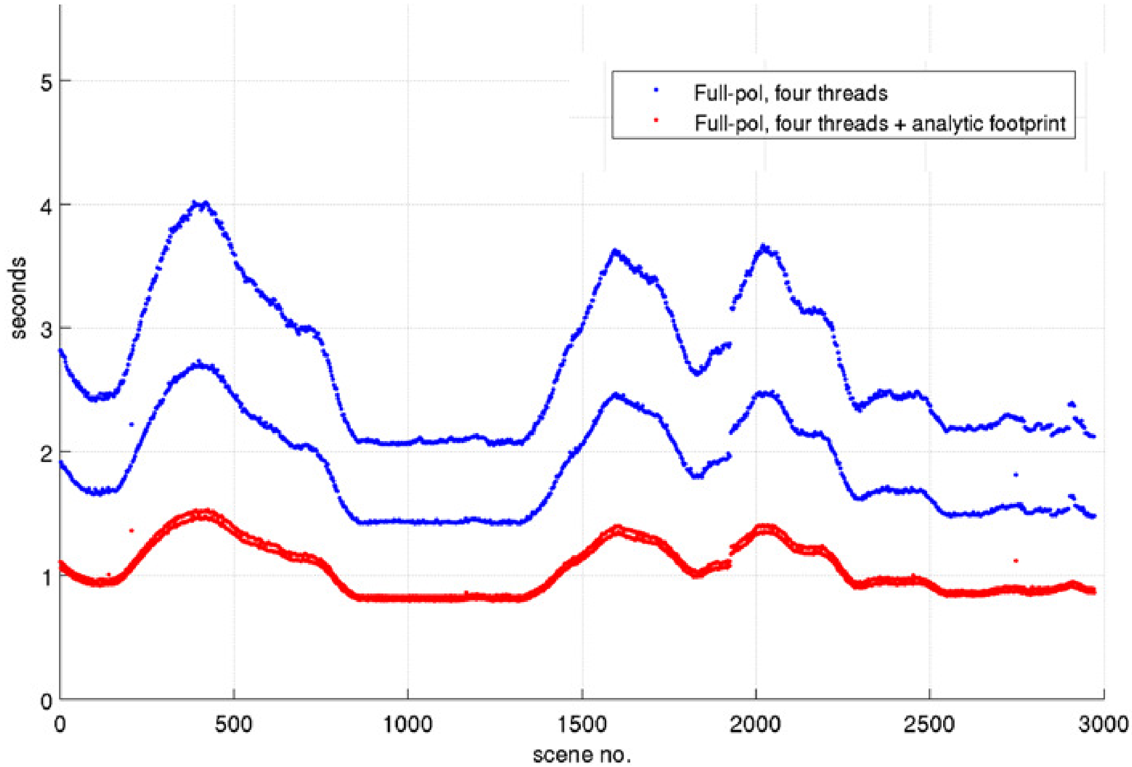



*Figure 8: Comparison of processing time in L1c.*

# 5. OPEN ISSUES

- There is still one small open issue that is to identify the scenes where the differences between the semi-axes stand out from the average are still to be identified and, if necessary, some corrections will be applied.