SMOS L1 Prototype

HW/SW Performance report

Code : SO-TN-DME-L1PP-0019
Issue : 2.1
Date : 17/07/07

	Name	Function	Signature
Prepared by	J. Freitas	Project Engineer	
	M. Ventura	Project Engineer	
	A. Guitierrez	Project Engineer	
Checked by	J. Freitas	Quality A. Manager	
Approved by	J. Barbosa	Project Manager	

DEIMOS Engenharia
Av. D. João II, Lote 1.17, Torre Zen, 10º
1998-023 Lisboa, PORTUGAL
Tel: +351 21 893 3017
Fax: +351 21 896 9099
E-mail: mailto:deimos@deimos.com.pt

This page intentionally left blank

Document Information

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

Internal Distribution		
Name	Unit	Copies

External Distribution		
Name	Organisation	Copies
Michele Zundo	ESA	1
Josep Closa	EADS CASA Espacio	1

Archiving	
Word Processor:	MS Word 2000
File Name:	SO-TN-DME-L1PP-0019-HW-SW-Performace-Report.doc
Archive Code:	P/STR/DME/03/013-038

Document Status Log

Issue	Change description	Date	Approved
0.1	Draft version containing preliminary results	17-01-2006	
1.0	Version 1.0 delivered after the OSAT	20-07-2006	
2.0	Version 2.0 for Phase 3 V2R	24-11-2006	
2.1	Prepared for Phase 3 V3.5	17-07-2007	

Table of Contents

List of Tables

1. INTRODUCTION

1.1. Purpose and Scope

The purpose of this document is to present the performance figures of the SMOS Level 1 Processor Prototype (L1PP) v1.3.5 in terms of processing time and input/output time. The main objective is to identify the performance bottlenecks of the L1PP and present possible mitigation mechanisms.

1.2. Acronyms and Abbreviations

For the list of acronyms, please refer to the "Directory of Acronyms and abbreviations" [RD.1].

1.3. Applicable and Reference Documents

1.3.1. Applicable Documents

Ref.	Code	Title	Issue
AD.1	SO-SOW-CASA-PLM-0855	Level 1 Processor Prototype Development Phase 3 and Support Activities. Statement of Work	01
AD.2	ECSS-E-40B	ECSS E-40 Software Engineering Standards	
AD.3	SO-DS-DME-L1PP-0006	SMOS L1PP System Concept	2.7
AD.4	SO-DS-DME-L1PP-0007	SMOS L1PP DPM L1a	2.3
AD.5	SO-DS-DME-L1PP-0008	SMOS L1PP DPM L1b	2.3
AD.6	SO-DS-DME-L1PP-0009	SMOS L1PP DPM L1c	2.1
AD.7	SO-DS-DME-L1PP-0012	SMOS L1 Processor Prototype ADD	1.3
AD.8	SMOS-DMS-TN-3400	SMOS L1 User/System Requirement Document	1.1
AD.9	SO-TR-DME-L1PP-0018	SMOS L1 Verification and Validation Plan	2.2
AD.10	SO-TDD-DME-L1PP-0022	SMOS L1 Prototype Test Data Description	2.2

Table 1: Applicable Documents

1.3.2. Reference Documents

Ref.	Code	Title	Issue
RD.1	SO-LI-CASA-PLM-0094	Directory of Acronyms and abbreviations	

Table 2: Reference Documents

2. OVERVIEW

The document is organized as follows:

- Section 3 describes the objectives, the approach and the methodology followed for doing the performance analysis. It also describes the Hardware and Software environment;
- Section 4 describes the different tests performed in this study;
- Section 5 shows the results of the different tests and identifies performance bottlenecks and possible improvements;
- Section 6 presents the conclusions based on the results obtained.

3. PERFORMANCE ANALYSIS OBJECTIVES AND APPROACH

The main objective of this study is to analyse the performance of L1PP in order to allow identifying bottlenecks of the prototype and issues that are candidates for an optimisation effort.

The idea is, in a first approach, to simply analyse the logs produced by the L1PP Orchestrator identifying the time spent in each processing unit as well as the time spent in Input/Output Operations[1]. The second step is to ignore the I/O output operations and run the prototype with a profiling tool in order to have statistics of the CPU Time consumed by the different functions.

For the profiling analysis, the open source *gprof* tool was used. Although presenting complete statistics, this tool presents a drawback: it only performs correctly on libraries statically linked and compiled with *–pg* flag. Since most of the libraries were initially linked dynamically, this led to an additional effort for compiling both the L1PP and external libraries appropriately.

The following tables describe the environment used to produce the results of section 5:

Table 3: Test Machine Earth

	Test Machines
Hardware Platform	AMD 64 dual processor @2.2GHz , 1MB L2 cache, 8Gb RAM - IBM server machine, **(Earth)** Xeon64 dual processor Dual-Core @2.0GHz , 4MB L2 cache, 8Gb RAM - DELL Precision machine, 8Gb RAM **(Earth2)**
Operating System	RedHat Enterprise Linux WS 4 update 4 for AMD64 and Intel EM64

Table 4: Software Environment

Level 1 Prototype	L1PP	1.3.5
Compiler	GNU gcc	4.2
Profiling Tool	gprof	2.15.92
Libraries	Java J2SE SDK	1.5
	Earth Explorer CFI	3.5
	Libxml2	2.6.16
	Binxml-fh	3.3
	BinX	1.2.2
	Xerces	2.7
	log4c	1.0.12
	lapack	3.0
	blas	(version included in lapack 3.0)
	fftw	3.1.2
	ATLAS	3.6.0

[1] The Input/Output Operations time includes all the time spent on the functions whose objective is to read or write Auxiliary Data Files (ADFs) or Products. This figure includes the access time to disk plus the processing time of the read/write functions.

4. TEST SCENARIOS DESCRIPTION

Three of the tests executed correspond to a nominal processing of dual and full polarisation products from L0 to L1c and three different data sets were used in the study: two 20 scenes data set (one in dual and the other in full polarization) used for profiling; one 2500 scenes data set used for overall performance assessment; and a FWAS product used for the generation of G and J+ matrices. Each one of the 20 scenes data set was run using Ideal and Real reconstruction.

An extra analysis has been performed on the last L1C product from the 2500 scenes scenario (contains only 73 scenes), in order to obtain a better insight of the time spent in each one of the Geolocation processing steps.

Each test has been run in a 64-bits baseline of L1PP version 1.3.5 and it is summarized in the following table.

Table 5 - Test Scenarios Description

Test	*Description*	*Note*
Test 1	Data Set with 2500 scenes run with optimisation (-O2 and ATLAS[2])	test-1440
Test 2	Data Set with 20 scenes in dual polarization run with profiling (-pg)	test-1110
Test 3	Data Set with 20 scenes in full polarization run with profiling (-pg)	test-1120
Test 4	G and J+ generation using single threaded and multi-threaded libraries	test-1401

[2] ATLAS is an Automatically Tuned Linear Algebra Software that provides a parallelized and optimized version of LAPACK and BLAS libraries.

5. TESTS RESULTS

This section presents the results of the tests execution. For each test, the following statistics are presented:

- ***Elapsed Time Results*** – time spent for executing each one of the modules. These results are extracted from the logs and correspond to the elapsed time (different from CPU time[3]). The following information is provided for each processing unit:

    - Number of Data Set Records: Number of records processed by processing unit;
    - Execution Time: Total execution time for each processing unit;
    - Execution Time without I/O: Total execution time without considering reading and writing products time (for each processing unit);
    - Percentage of I/O time: Percentage of time spent reading and writing the products.

  In addition to the statistics per processing unit, the figures are also presented for the overall total time.

- ***CPU Time results per function*** – output of the profiler tool. The following information is provided for the functions that consume more CPU Time:

    - Function Name: Name of the function to each the statistics refer to;
    - % of Total CPU Time: Percentage of the total (from L0 to L1c) CPU time spent by the function;
    - Number of Calls: Number of times that the function is called;
    - Self Seconds: Number of CPU seconds spent exclusively on the function;
    - Cumulative Seconds: This is the cumulative total number of seconds the computer spent executing this functions, plus the time spent in all the functions above this one in the results table;
    - Library: Identification the library that contains the function.

- ***CPU Time results per libraries*** – In addition to the previous statistics presented by function, similar statistics are presented but considering a decomposition by library.

[3] The elapsed time is the difference between the system time at the end of the processing and at the beginning. This time is influenced by other processes that may be running on the test machine. CPU Time corresponds to CPU processing time actually allocated to the test process, being independent from other processes that may be running on the test machine.

A short analysis of the results is also provided for each test.

5.1. Test 01

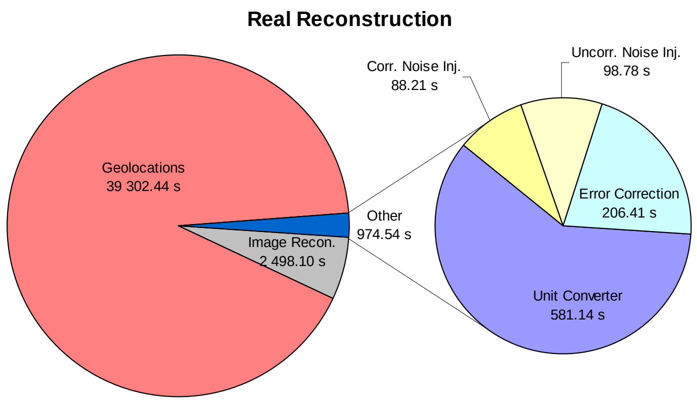
Test 1 has the following characteristics:

- Data Set: Half orbit (2500 scenes);
- Compilation Flags: Optimisation (-O2 with parallelized libraries - ATLAS);
- Test processing time: ≈ 11 hours and 54 minutes on Earth test machine.

Table 6: Elapsed Time per Processing unit

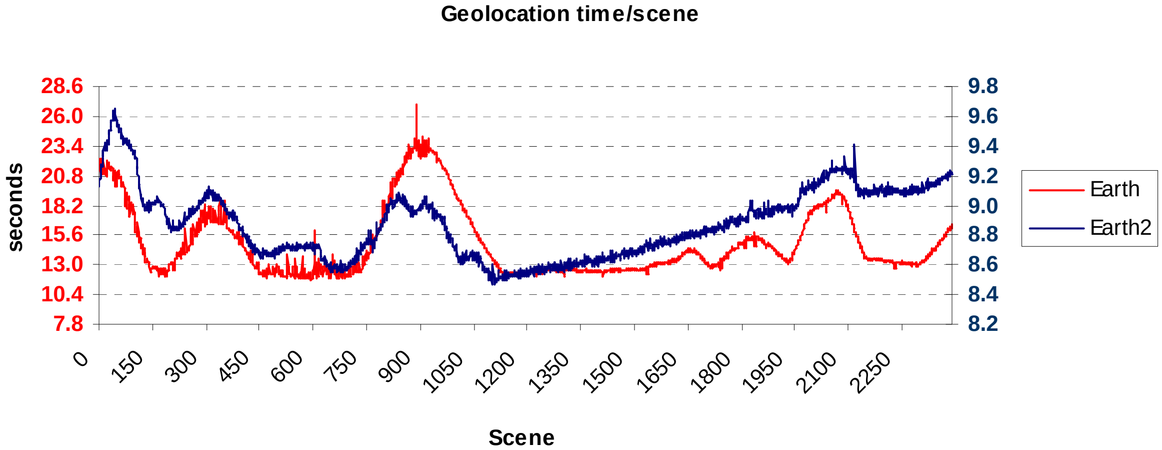
Processing Unit	Time	time/scene
Unit Converter	581.14 s	0.23 s
Corr. Noise Inj.	88.21 s	3.68 s
Uncorr. Noise Inj.	98.78 s	9.88 s
Error Correction	206.41 s	0.08 s
Image Recon.	2 498.10 s	1.01 s
Geolocations	39 302.44 s	15.96 s

Figure 1: Pie diagrams representing the time taken by each Processing Unit



From the previous table and figure, it is clear that the Geolocation is the most time consuming processing unit, together with the Image Reconstruction, of the overall scenario processing time.

Figure 2: Geolocation time per scene run on each SMOS test machines

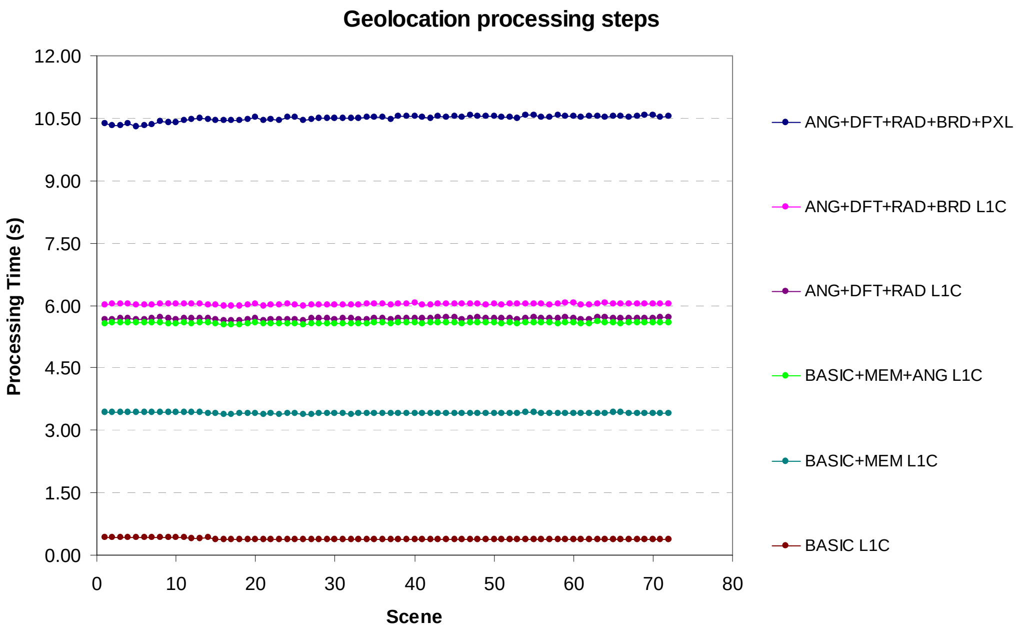


It is quite clear that Earth2 machine (using the right YY scale) is much faster than Earth. It is, however, possible to identify a similar behaviour while processing a group of snapshots that cause the time to increase. It might be caused by the search in the Digital Global Grid.

5.1.1. Geolocation analysis

In order to gain a better insight on the Geolocation processing time, an extra analysis has been performed on the last L1C product containing 73 scenes. The following results were obtained on Earth2 test machine.

Figure 3: Time spent on the different Geolocation steps.



- The lowest line (brown) represents the "unavoidable" computations: computation of the EAF-FOV, its projection on the ground and retrieval of the DGG pixel list falling inside.
- The next line (teal) considers the memory allocations needed to store the pixel values once they are computed, but no other calculations.
- The next line (green) represents the first level of computations needed for each pixel processing, namely the computation of all related angles (incidence, azimuth, S/C to pixel, pixel to S/C…) and the unit circle coordinates (xi-eta).
- The line above (red) adds only the Discrete Fourier Transform of the L1b data plus the radiometric accuracy computation, which you can see is just a small contribution to the overall processing. The DFT is composed by one "for loop", which is one of the things best optimised by the compilers.
- The next line (pink) includes the computation of pixel flags, including the flagging of pixels too close to the border of the EAF-FOV.
- Finally the last line (blue) represents the complete L1c processing, and the quantitative jump in processing time is due to the computation of the pixel footprint.

5.2. Test 02

Test 2 has the following characteristics:

- Data Set: 20 scenes in dual polarization;
- Compilation Flags: -pg;

- Test processing time: ≈ 10 minutes for each reconstruction method on Earth test machine.

The objective of this test was to profile L1PP and identify the performance bottlenecks of the prototype.

The following table presents the functions that consume more CPU time through an L0 to L1c execution:

Table 7: CPU Time per function in Real reconstruction.

Function Name	% of Total CPU Time	Cumulative Seconds	Self Seconds	Number of Calls	Library
xp_target_inter	12.4	46.17	46.17	922052	CFI
sin	10.56	85.47	39.3		Math
cos	9.99	122.65	37.18		Math
geolocation	6.4	146.48	23.83	1	Geolocation
dgemm_	4.31	162.53	16.05	1220572	LAPACK
zmulz	2.89	173.31	10.78	426011463	Core Components
__pthread_alt_unlock	2.85	183.93	10.63		
zmuld	2.44	193.02	9.09	425701845	Core Components
checkAfFovBorder	2.29	201.55	8.53	152542	Geolocation
read	1.97	208.89	7.34		
_int_malloc	1.86	215.81	6.92		
computeUV	1.49	221.36	5.55	55820	Image Reconstruction
__pthread_alt_lock	1.27	226.07	4.71		
dgemv_	1.26	230.75	4.68	2116559	LAPACK
malloc_consolidate	1.24	235.36	4.61		
conjz	1.17	239.72	4.37	212898330	Core Components
BxDataObject::BxDataObject(BxDataObject const&)	1.1	243.82	4.1	36992984	BinX
__printf_fp	1.04	247.7	3.88		
__profile_frequency	1.03	251.52	3.82		

Table 8: CPU Time per function in Ideal reconstruction.

Function Name	% of Total CPU Time	Cumulative Seconds	Self Seconds	Number of Calls	Library
xp_target_inter	11.15	45.15	45.15	922052	CFI
sin	9.68	84.37	39.22		Math
cos	9.43	122.59	38.22		Math
computeUV	7.62	153.46	30.87	311558	Image Reconstruction
geolocation	5.97	177.66	24.2	1	Geolocation
dgemm_	4.02	193.93	16.27	1220572	LAPACK
zmulz	2.82	205.37	11.44	426011463	Core Components
__pthread_alt_unlock	2.72	216.39	11.02		
zmuld	2.1	224.91	8.52	425701845	Core Components
checkAfFovBorder	1.94	232.76	7.85	152542	Geolocation
_int_malloc	1.81	240.08	7.32		

Function Name	% of Total CPU Time	Cumulative Seconds	Self Seconds	Number of Calls	Library
__printf_fp	1.69	246.92	6.84		
read	1.65	253.59	6.67		
__pthread_alt_lock	1.28	258.78	5.2		
malloc_consolidate	1.22	263.73	4.95		
conjz	1.2	268.59	4.86	212898330	Core Components
BxDataObject::BxDataObject(BxDataObject const&)	1.07	272.92	4.33	37517208	BinX
__profile_frequency	1	276.98	4.06		

Table 9 - Test 2 - CPU Time per library

Function Name	% of Total CPU Time
CFI	11 ~ 12
LAPACK	4 ~ 6
Geolocation	8 ~ 9
Core Components	6 ~ 7
Image Reconstruction	2 ~ 8

From the analysis of the profiling the following conclusion can be extracted:

- Earth Explorer CFI library functions consume around 12% of the total CPU time of L1PP. This is something that can not be mitigated, as there is a large amount of computations that need to be performed on a pixel basis during the geolocation. After analysing the algorithms and the code it was concluded that a reduction of the number of calls to EE CFI functions is not feasible. The only way of improving the performances of these computations is to use a multithread approach in geolocation[4], considering the number of snapshots that have to be processed and separate the non-overlapping group of snapshots into different cycles (one per processor).

- As seen by the statistics, the matrix vector multiplications (function *dgemv* and *dgemm* from Lapack libraries) performed in L1b both in the Foreign Sources removal and in the Image Reconstruction are very consuming in terms of CPU time. The usage of these functions may be optimised in two ways:

    - Matrix vectors multiplications are used for multiplying G and J+ matrices by vectors that are computed for each scene. A way of improving the performances is to process several scenes in a single matrix-matrix multiplications reducing the number of operations;

    - Optimisation may be achieved by parallelising the matrix-vector or matrix-matrix operations. LAPACK library provided in ATLAS already supports parallelization of many functions and the effort needed for implementing this feature is reduced. [See 5.4]

[4] The impact of the implementation of multithread approach in Earth Explorer CFI functions was not analysed at this stage.

- Regarding the L1PP internal functions, the analysis of the profiling results allowed to identify possible performance improvement strategies:

Table 10: Performance Improvement suggestions

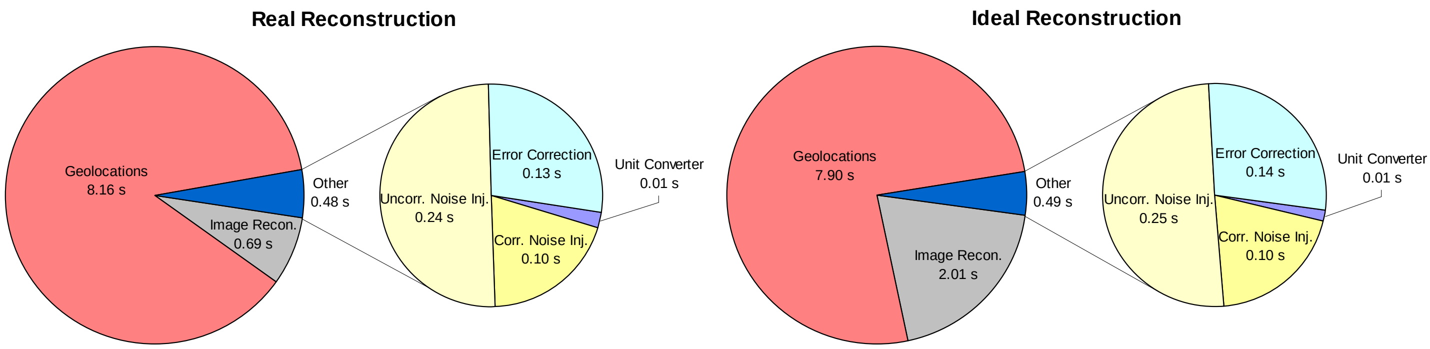
Function	Library	Description	Optimisation Suggestion
zmulz zmuld conjz complexd	Core Components	Generic functions used for performing basic complex numbers operations	Since these functions are called a large number of times it would be a good idea to use the Complex GCC Extension that has proved to have a very good performance.

Comparing both reconstruction methods:

Table 11: Elapsed Time per Processing Unit

Processing Unit	Real Reconstruction		Ideal Reconstruction		1-real/ideal
	Overall	time/scene	Overall	time/scene	
Unit Converter	0.54 s	0.01 s	0.46 s	0.01 s	-17%
Corr. Noise Inj.	2.30 s	0.10 s	2.32 s	0.10 s	1%
Uncorr. Noise Inj.	2.41 s	0.24 s	2.48 s	0.25 s	3%
Error Correction	2.66 s	0.13 s	2.74 s	0.14 s	3%
Image Recon.	**13.75 s**	**0.69 s**	**40.29 s**	**2.01 s**	**66%**
Geolocations	163.19 s	8.16 s	157.91 s	7.90 s	-3%

Figure 4: Pie diagrams representing the time taken by each Processing Unit



5.3. Test 03

Test 3 has the following characteristics:

- Data Set: 20 scenes in full polarization;
- Compilation Flags: -pg;

- Test processing time: ≈ 12 minutes for each reconstruction method on Earth test machine.

The objective of this test was to profile L1PP and identify the performance bottlenecks of the prototype.

The following table presents the functions that consume more CPU time through an L0 to L1c execution:

Table 12: CPU Time per function in Real reconstruction.

Function Name	% of Total CPU Time	Cumulative Seconds	Self Seconds	Number of Calls	Library
xp_target_inter	14.48	70.87	70.87	1291464	CFI
sin	10.98	124.59	53.72		Math
cos	10.59	176.42	51.83		Math
geolocation	6.6	208.7	32.28	1	Geolocation
dgemm_	3.77	227.13	18.43	1709518	LAPACK
zmulz	3.14	242.47	15.34	596579913	Core Components
__pthread_alt_unlock	2.72	255.77	13.3		
read	2.57	268.33	12.56		
zmuld	2.42	280.19	11.86	596323305	Core Components
checkAfFovBorder	2.21	291.02	10.83	213660	Geolocation
_int_malloc	1.72	299.45	8.43		
computeUV	1.69	307.74	8.29	83730	Image Reconstruction
dgemv_	1.52	315.2	7.46	2964150	LAPACK
__pthread_alt_lock	1.19	321.03	5.84		
malloc_consolidate	1.19	326.85	5.82		
__profile_frequency	1.01	331.77	4.92		

Table 13: CPU Time per function in Ideal reconstruction.

Function Name	% of Total CPU Time	Cumulative Seconds	Self Seconds	Number of Calls	Library
xp_target_inter	14.21	74.83	74.83	1291464	CFI
sin	10.45	129.9	55.07		Math
cos	10.32	184.25	54.35		Math
geolocation	6.16	216.71	32.46	1	Geolocation
computeUV	5.73	246.91	30.2	302994	Image Reconstruction
dgemm_	3.32	264.42	17.51	1709518	LAPACK
zmulz	2.93	279.87	15.45	596579913	Core Components
__pthread_alt_unlock	2.59	293.52	13.65		
zmuld	2.2	305.1	11.59	596323305	Core Components
checkAfFovBorder	2.12	316.26	11.16	213660	Geolocation
_int_malloc	1.77	325.56	9.3		
read	1.72	334.63	9.07		
malloc_consolidate	1.42	342.13	7.5		
__printf_fp	1.3	348.99	6.86		
__pthread_alt_lock	1.26	355.65	6.66		
__profile_frequency	1	360.92	5.27		

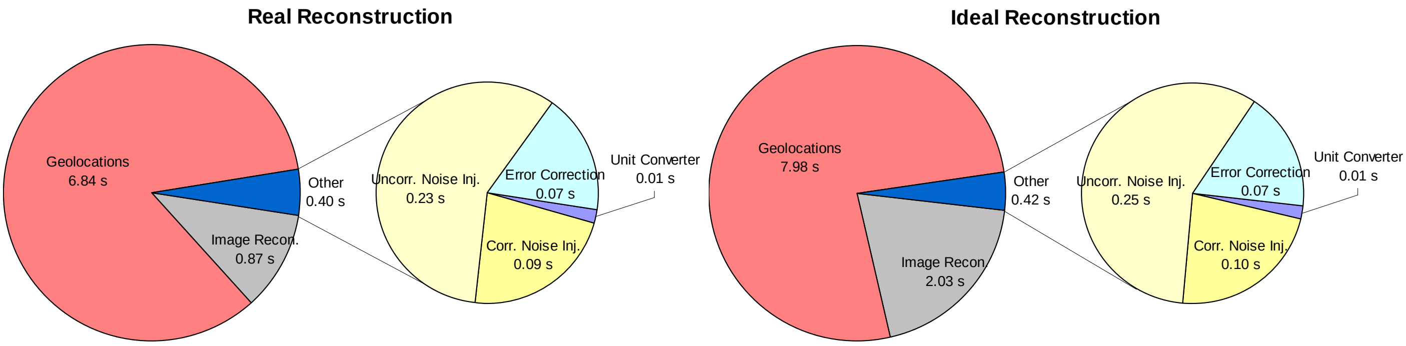
Similar conclusions to the Test 02 can be drawn from this test on what refers to the most time consuming Processing Units and the improvements that can be performed.

Let's compare both reconstruction methods.

Table 14: Elapsed Time per Processing Unit

Processing Unit	Real Reconstruction		Ideal Reconstruction		1-real/ideal
	Overall	time/scene	Overall	time/scene	
Unit Converter	0.43 s	0.01 s	0.44 s	0.01 s	2%
Corr. Noise Inj.	2.13 s	0.09 s	2.28 s	0.10 s	7%
Uncorr. Noise Inj.	2.34 s	0.23 s	2.46 s	0.25 s	5%
Error Correction	2.79 s	0.07 s	2.97 s	0.07 s	6%
Image Recon.	**17.41 s**	**0.87 s**	**40.54 s**	**2.03 s**	**57%**
Geolocations	191.54 s	6.84 s	223.31 s	7.98 s	14%

Figure 5: Pie diagrams representing the time taken by each Processing Unit



5.4. Test 04

Test 3 has the following characteristics:

- Data Set: Generation of G and J+ matrices;
- Compilation Flags: -O2 using Single Threaded and Parallelized[5] versions of LAPACK and BLAS libraries.
- Test machine: Earth

BLAS and LAPACK libraries are the core libraries used in the prototype to perform matrix and vector manipulation. In case of J+ generation those libraries are used to decompose and invert the G matrices.

[5] Parallelized version of the libraries provided by ATLAS

This test was executed using different compilations of L1PP; one using single threaded LAPACK and BLAS libraries and the other one using the Parallelized version of the same libraries.

The following results were obtained:

Table 15 - Time of G and J Matrices generation

Description		Single thread	Multi thread	Gain
UPC Unit Circle G Matrix	Load Data	0:04:21	0:04:00	
	Generation	0:05:11	0:03:18	
	Overall	0:09:31	0:07:18	23%
UPC Hexagon G Matrix	Load Data	0:04:20	0:04:02	
	Generation	0:05:10	0:03:13	
	Overall	0:09:30	0:07:14	24%
J Matrix	FFT preparation	0:00:11	0:00:11	
	H part generation	0:15:08	0:17:43	
	Invert H part	0:57:02	0:23:06	
	write to file	0:00:02	0:00:02	
	V part generation	0:14:58	0:17:11	
	Invert V part	0:54:45	0:24:10	
	write to file	0:00:02	0:00:02	
	FFT preparation	0:00:23	0:00:21	
	HV part generation	1:42:43	1:37:35	
	Invert HV part	6:39:54	3:33:39	
	write to file	0:00:22	0:00:12	
	Overall	**10:45:29**	**6:34:13**	39%

Because the libraries are parallelized, the prototype can use the multiprocessor capabilities of the test machine, which in this case is translated in an overall gain of around 25% for the G matrix generation and 40% for the J+ matrix generation. Note that the parallelization isn't meant to affect the I/O functions and the differences of time noticed are only caused by the availability of the disc at runtime.

6. CONCLUSIONS

As a conclusion, and as mentioned previously, the main bottleneck of the prototype at this moment is the high number of computations that must be performed for each pixel, requiring also a high number of calls to the EE CFI libraries. Geolocation module processing represents around 80% of the total processing time of the L1PP and is a good candidate module for an additional optimisation effort, but, as seen in 5.1.1, its optimization may not be trivial.

From the tests performed in 5.2 and 5.3, it is clear that the usage of Real Reconstruction method in Image Reconstruction processing unit improves around 60% the processing time.

Some of the suggestions for improving the performance of the L1PP presented in previous versions of this report were already implemented with good results. Additional suggestions are presented in this version and will be analysed at a later stage.

Considering the current state of the Prototype implementation, the average processing rate is around 17 seconds per scene.