Test procedure for the RMIB Imager Process - Version 3 of the implementation

Nicolas Clerbaux

September 18, 2000

Contents

CHANGE RECORD

Issue	Date	Approved by	Reason for change
Draft	01/08/00		new document

1 Introduction

1.1 Purpose of this document

Description of the test procedure that is used to validate the set of programs that form the **imager processing** (also called **SEVIRI processing**) within the general GERB processing at RMIB. These tests refer to the version 3 of the imager processing implementation (the latest version). This version is planned to be used for the operational processing of the SEVIRI data after the launch of MSG-1.

1.2 Scope of this document

This document only described the test procedure. The choice of the methods are decribed in the RMIB technical notes [6] (spectral modelling), [5] (scene identification), [4] (angular modelling). The implementation of the imager processing version 2 is described in [1]. Note that to date there is no finalised documentation for the version 3 of the imager processing.

1.3 Dates of the tests

Most of the tests described in this document have been performed during the development of the imager processing programs. Nevertheless, the test procedure described here has been defined during the period July 15^{th} to 31^{th} 2000. During this period, all the tests described have been done again using the latest version of the implementation (known as Version 3).

1.4 Imager processing overview

Programs sequence

The imager processing includes 8 successive programs. These programs are executed for each imager slot in the order given by the following table:

name	object
imager_common_proc	pre-processing of the SEVIRI L1.5 data
imager_spectralmod	spectral modelling
imager_cloudanalysis	cloud analysis
imager_rad2fluxes	angular conversion
barg_proc	BARG processing
imager_archive	archiving into HDF files + internal archive
imager_ftp	put interesting data into our FTP site
imager_clean	remove temporary data

This document deals with the 4 first programs in the table herebefore. The test procedure for the **barg_proc, imager_archive, imager_ftp** and **imager_clean** programs is not documented here because these programs are regularly modified and will not be used as it for the operational processing at RMIB.

imager_common_proc

This first program accesses the imager level 1.5 data (SEVIRI or Meteosat count images) and derives some intermediate images that are needed for the following processings. Note that all the accesses at imager data are done by this processing.

imager_spectralmod

This second program estimates the unfiltered solar and thermal radiances and the GERB shortwave and longwave radiances from the set of NB radiances of the imager.

imager_cloudanalysis

This third program characterises the cloud cover within the Field Of View (FOV) of the imager.

imager_rad2fluxes

In a first step this program selects the best suited Angular Dependency Models (ADM's) and then applies these models to convert the unfiltered solar and thermal radiances into the corresponding fluxes.

barg_proc

This program performs the estimate of the BARG (Binned Averaged Rectified Grid) products from the SHI (Standard High Resolution) products.

imager_archive

This program creates a set of HDF files that contain the SHI and BARG products.

imager_ftp

This program performs the copy of the HDF files on our FTP site for external use.

imager_clean

Remove all the temporary data (all the data excepted the archived ones). The original input imager data are also removed in case of successful processing.

1.5 General testing procedure

Tests using development programs

Each time it was possible, the output of the **operational programs** have been compared to the output of smaller **development programs**. These development programs usually perform only a small part of the processing such as: transforming a radiance image into a reflectance image, conversion of a radiance image into brightness temperature image, ...

This kind of tests allows the detection of some mistakes when implementing the operational version. Nevertheless, these 2 kinds of programs often use some common routines from a shared C-library (libbats.a) and then separate tests must be performed for these routines.

Image visualisation tests

Each time it was possible, a visualisation test has been done. These test consist simply to visualised an intermediate or final image using the **viewrma** program. Instead most of the current visualisation program (xv, ...), the viewrma program allows the visualisation of floating point images (nearly all our images are floating point).

The image visualisation tests allow the detection of a lot of potential mistakes of implementation such as (for example) the use of bad units, ...

Tests on various hardware

The tests described in this document have been performed on various computing platforms. This allows the detection of some problems related to system calls (date and time representation) or number representation. The results concerning the execution time are the ones obtained using our **tornado** workstation (DEC alpha family) which is the platform planned to be used for the operational GERB processing.

Use of Meteosat-7 data

Some artificial SEVIRI inputs images have been generated for testing purposes (see after). Nevertheless, these images are always filled with constant count values. These artificial SEVIRI data allow to test the correct execution of the programs and also the estimation of needed resources (computation time and memory space).

Additional tests have been done using as input data the images acquired by the imager on Meteosat-7 (the current EUMETSAT geostationary satellite at 0^o longitude). These Meteosat slots are mainly used to test the cloud identification results.

RMIB name	channel	value $[Wm^{-2}sr^{-1}]$	count
S1	0.6μ	18.423	512
S2	0.8μ	9.789	512
S3	1.6μ	4.709	512
T1	6.2μ	3.363	512
T2	7.3μ	2.510	512
T3	8.7μ	2.058	512
T4	9.7μ	1.506	512
T5	10.8μ	5.572	512
T6	12.0μ	4.920	512
T7	13.4μ	5.698	512

Table 1: Typical radiances and count values for the SEVIRI channels.

1.6 Input data for the tests

Introduction

The basic input data for the imager processing are SEVIRI count images (also named SEVIRI level 1.5). To tests the imager processing programs, 3 sets of SEVIRI-like images have been generated. A typical values for the count in the SEVIRI images is 512, in the middle of the dynamic range [0,1024]. The table 1 gives the corresponding radiances using a provisional calibration for the SEVIRI channels (this calibration is described in [3]). As described before, 2 slots of Meteosat-7 images are also used.

SEVIRI test slot 1 (20001851800*.rma)

This slot correspond to images taken at 18 p.m. for the Julian day 185 of 2000. All the count images are filled with 0 values. This test slot is useful to test the viewing angle part of the processing (imager_common_proc).

SEVIRI test slot 2 (20000012400*.rma)

This slot correspond to images taken at midnight for the Julian day 1 of 2000. For this slot, only the thermal channels images are generated. The visible channels do not exist. All the thermal channels count images are filled with the values given in table 1. This test slot is useful to test that the visible part of the processing is skipped when there is no visible band input images.

SEVIRI test slot 3 (20000011200*.rma)

This slot correspond to images taken at noon for the Julian day 1 of 2000. The NB count images are filled with the values given in table 1.

Meteosat-7 test slot A

The letter A refers to the EUMETSAT format A geometry (full disk). We selected the 20002051500 slot for the tests (that corresponds to Julian day 205 of the year 2000, 15h00'). Note that, as the WV image is not received at 15h00', the image at 15h30' is used.

archive name	band
20002051500aivhir010.pgm.gz	visible (S1)
20002051530awwv014.pgm.gz	water vapour (T1)
20002051500aivhir010.pgm.gz	infrared (T2)

Meteosat-7 test slot B

The letter B refers to the EUMETSAT format B geometry (Europa, Atlantic and North Africa). We selected the 20002061000 slot for the tests (that corresponds to Julian day 206 of the year 2000, 10h00').

archive name	band
20002061000bivhir010.pgm.gz	visible (S1)
20002061000bwwv014.pgm.gz	water vapour (T1)
20002061000bivhir010.pgm.gz	infrared (T2)

Directories

- these input data are stored in the directory: **planet:/usr/people/nic/Imagerprocess/Test_V3/Count**
- the batch file **create_data.bat** (located in the same directory) can be used to generate these 3 sets of SEVIRI images.

1.7 Configuration Data

Introduction

The input constant data for the imager processing programs are all stored in configuration files. These files are:

- planet:/usr/people/nic/ImagerProcess/Data/Config/seviri_cfg.txt
- planet:/usr/people/nic/ImagerProcess/Data/Config/ms7_A_cfg.txt
- planet:/usr/people/nic/ImagerProcess/Data/Config/ms7_B_cfg.txt

Imager calibration

Provisional values for the conversion of SEVIRI counts into radiance are used. These values are described in the RMIB internal document [3]. An offset value of 10 is used for each channels. The calibration of the Meteosat-7 imager is given on the EUMETSAT web site.

Imager geolocation

The geolocation tables for the SEVIRI and Meteosat-7 imager have been created using the **create_geoloctable** program. These tables are stored in the directory:

- planet:/gerbdata/nic/ImagerProcess/Geolocation

1.8 Table of image's suffixes

The tables hereafter gives the suffixes used to store the intermediate and final images on the disk. The complete names are formed by addition of a prefix that identifies the slot (YYYYDDDHHMM). In the table, the **X** is the imager channel identificator. For the resolution column, the **F** means a full resolution image, the **S** a subsampled resolution image and the **BARG** a product at the GERB-like resolution..

SHI products

suffix	res.	program	image's content
_count_SX.rma	S	imager_common_proc	subsampled count
_count_TX.rma	S	imager_common_proc	subsampled count
_rad_SX.rma	S	imager_common_proc	subsampled radiance
_rad_TX.rma	S	imager_common_proc	subsampled radiance
_vza.rma	S	imager_common_proc	subsampled viewing zenith angle
_sza.rma	S	imager_common_proc	subsampled solar zenith angle
_ra.rma	S	imager_common_proc	subsampled relative azimuth angle
_ref_SX.rma	F	imager_common_proc	reflectance for solar channels
_bt_TX.rma	F	imager_common_proc	brightness temperature for thermal channel
_sol_rad.rma	S	imager_spectralmod	unfiltered solar radiance
_th_rad.rma	S	imager_spectralmod	unfiltered thermal radiance
_sw_rad.rma	S	imager_spectralmod	filtered GERB shortwave radiance
_lw_rad.rma	S	imager_spectralmod	filtered GERB longwave radiance
_ca_flag.rma	S	imager_cloudanalysis	internal RMIB cloud flag
_ca_fraction.rma	S	imager_cloudanalysis	cloud fraction (in ninth)
_ca_C.rma	S	imager_cloudanalysis	mean C for cloudy pixels
_ca_tau.rma	S	imager_cloudanalysis	mean τ for cloudy pixels
_ca_phase.rma	S	imager_cloudanalysis	mean phase for cloudy pixels
_sol_flux.rma	S	imager_rad2fluxes	unfiltered solar flux
_th_flux.rma	S	imager_rad2fluxes	unfiltered thermal flux

BARG products

suffix	res.	program	image's content
_vza_barg.rma	BARG	barg_proc	subsampled viewing zenith angle
_sza_barg.rma	BARG	barg_proc	subsampled solar zenith angle
_ra_barg.rma	BARG	barg_proc	subsampled relative azimuth angle
_sol_rad_barg.rma	BARG	barg_proc	unfiltered solar radiance
_th_rad_barg.rma	BARG	barg_proc	unfiltered thermal radiance
_ca_fraction_barg.rma	BARG	barg_proc	cloud fraction (in ninth)
_ca_C_barg.rma	BARG	barg_proc	mean C for cloudy pixels
_ca_tau_barg.rma	BARG	barg_proc	mean τ for cloudy pixels
_ca_phase_barg.rma	BARG	barg_proc	mean phase for cloudy pixels
_sol_flux_barg.rma	BARG	barg_proc	unfiltered solar flux
_th_flux_barg.rma	BARG	barg_proc	unfiltered thermal flux

2 imager_common_proc

2.1 Program Description

Object of the program

This program derives from the imager level 1.5 data (**count images**) the basic quantities that serve as input for the others programs of the imager processing. Within the imager processing programs, **imager_common_proc** is the only one program that accesses directly the imager data.

This program performs:

- the calibration of the imager data (transformation from count value to floating point radiance),
- the computation of the viewing zenith,solar zenith and relative azimuth angle images,
- the subsampling of the count and radiance images,
- the estimation of the reflectance images (for solar channels) or the brightness temperature images (for thermal channels),
- the creation of an header file that can be accessed by the following programs.

Program Inputs / Output

inputs	**outputs**
configuration file	header file
SEVIRI count images	subsampled radiances images
calibration file	subsampled viewing angle images
acquisition schedule file	reflectance images (solar channels)
ρ computation schedule file	brightness temperature images (thermal channels)
imager geolocation files	subsampled count images

Operational program execution

The operational program has been executed on the 5 test slots described before. This execution is done using the commands:

- imager_common_proc 20001851800 seviri_cfg.txt -v
- imager_common_proc 20001851200 seviri_cfg.txt -v
- imager_common_proc 20001852400 seviri_cfg.txt -v
- imager_common_proc 20002051500 ms7_A_cfg.txt -v
- imager_common_proc 20002061000 ms7_B_cfg.txt -v

2.2 Execution tests - Passed

Purpose of the test

Check the good execution of the program and the resources needed for the execution. The execution test is done on various hardware platform (PC Linux, SGI origin, DEC Alpha).

Tests methodology - criteria

The good execution of the various steps of the processing has been checked for the 5 sets of imager data described before. During the execution, the informations displayed on the standard output (we used the verbose mode -v) are checked. It is also checked that all the expected output files have been generated.

Execution time

imaging device	total (s)	user (s)	system (s)
SEVIRI	48s	40s	6s
MS7-A	7s	6s	1s

The difference in computation time is due to the higher number of channels for SEVIRI (10 instead of 3 for MS7) and also the higher number of imager pixels ($13\,10^6$ instead of $6\,10^6$ for MS7-A).

2.3 Header file tests - Passed

Purpose of the test

Validate the creation and the content of the header files that are generated by the **imager_common_proc** operational program.

Tests methodology - criteria

Edition of the header files (which is an ASCII file) and check their content for the 5 test slots. In these header files, the **SOL_FLAG** value must always be 1 excepted for the SEVIRI test slot 2 for which a value of 0 (no visible band input) is expected.

2.4 Subsampled count images tests - Passed

Purpose of the test

Check the generated subsampled count images which are generated by the **imager_common_proc** operational program.

Tests methodology - criteria

1. The subsampled count images have been compared with images generated using the **im_divide_size** development programs. The corresponding images are compared using the utility program **imdiff** which compares values for 2 images. The images generated by the operational program and by the **im_divide_size** must be the same (same sizes and same values in the data field).
2. for the MS7 slots, the subsampled count images are visualised using the visualisation program **viewrma**. The subsampled images must look as the input count images but which a reduced spatial resolution.

Used development/utility programs

- **im_divide_size** : is used to perform the image subsampling,
- **imdiff** : is used to analyse the difference between two images.

2.5 Subsampled radiance images tests - Passed

Purpose of the test

Check the generated subsampled calibrated NB radiance images which are generated by the **imager_common_proc** operational program.

Tests methodology - criteria

3 tests must be done to check the validity of the subsampled radiance images generated by the **imager_common_proc** operational program:

1. These images have been compared with images generated using the **im_linear_transf** and **im_divide_size** development programs. The corresponding images are compared using the utilitary program **imdiff**. The two images must be the same (same sizes and same values in the data field).
2. The central values in the radiance images for the test slot 3 are checked using the utilitary program **im_info.** The values at the centre of the images must be the same as the ones given in table 1.
3. The NB radiance values should never be negative. This must be checked by image visualisation (viewrma) and inspection of the radiance value for the test slot MS7-A.

Used development/utilitary programs

- **im_linear_transf** : is used to perform the calibration of the imager count images,
- **im_divide_size** : is used to perform the image subsampling,
- **imdiff** : is used to analyse the difference between two images,
- **im_info** : is used to display some general information about an image (sizes, data type) and the value of the data field for the central pixel of this image.

2.6 Viewing angles images tests - Passed

Purpose of the tests

Check the generated subsampled viewing angle images which are generated by the **imager_common_proc** operational program.

Visualisation tests

The images of the viewing angles (viewing zenith, solar zenith and relative azimuth) generated by the **imager_common_proc** operational program are visualised using the **viewrma** visualisation program for the MS7-A and MS7-B test slots. It is checked:

- that the viewing zenith angle is null at the satellite subpoint,
- that the solar zenith angle is null at the sun's subpoint,
- that the relative azimuth image shows discontinuities for these two points,
- that the 3 angle image are expressed in radian and not in degrees.

Test using development programs

The images of the viewing angles have been compared with images generated using the **imager_acq_time** and **viewing_angle** development programs. The corresponding images are compared using the utilitary program **imdiff** that compares values for 2 images. These images must be the same (same sizes and same values in the data field).

This test is performed for:

- SEVIRI test slot 1 (18h00).
- the MS7-A and MS7-B test slots.

Used development programs

- **imager_acq_time** : create an ASCII file that contains the acquisition times for each line of the imager,
- **viewingangle** : create the 3 angle images (viewing zenith, solar zenith and relative azimuth),
- **imdiff** : is used to analyse the difference between two images.

2.7 Reflectance images - Passed

Purpose of the test

Check the reflectance images which are generated by the **imager_common_proc** operational program.

Visualisation tests

The images of the reflectance for the solar channels generated by the **imager_common_proc** operational program are visualised using the **viewrma** visualisation program for the MS7-A and MS7-B test slots. It is checked:

- the reflectance ρ varies in the range [0,1]. In particular ρ must never be negative.
- the reflectance for optically thick clouds is close to $\rho \sim 0.8$,
- the reflectance must be small on the ocean ($\rho \sim 0.1$),
- it is also checked that there is no discontinuities in the reflectance image introduced by the conversion formula.

Test using development programs

The images of the reflectance for the solar channels have been compared with images generated using the **rad2ref** development programs. The corresponding images are compared using the utilitary program **imdiff** that compares values for 2 images. The two images must be the same (same sizes and same values in the data field).

This test is performed for:

- SEVIRI test slot 1 and 3,
- the MS7 test slots.

Used development program

- **im_linear_transf** : is used to perform the calibration of the imager count images,
- **rad2ref** : compute the reflectance image from a radiance image.

2.8 Brightness temperature image - Passed

Purpose of the test

Check the brightness temperature image which is generated by the **imager_commom_proc** operational program.

Visualisation tests

The images of the brightness temperature (BT) for the thermal channels generated by the **imager_common_proc** operational program are visualised using the **viewrma** visualisation program for the MS7-A and MS7-B test slots. On these images is checked that the estimated brightnesss temperature is in accordance with the physical law:

- high level cloud must be cold ($T_B \sim 200K$),
- surface and low level clouds must be hot (up to $T_b \sim 320K$ for the Saharan desertic areas),
- it is also check that the unit used for the brightness temperature images is the Kelvin.

Test using development programs

The images of brightness temperature have been compared with images generated using the **bright_temp** development programs. The corresponding images are compared using the utilitary program **imdiff** that compares values for 2 images. The two images must be the same (same sizes and same values in the data field).

This test is performed for the :

- SEVIRI test slot 2,
- the MS7 test slots.

Used development program

- **im_linear_transf** : is used to perform the calibration of the imager count images,
- **bright_temp** : estimation of the brightness temperature image from a thermal radiance image using the EUMETSAT logarithmic formula.

3 imager_spectralmod

3.1 Program Description

Object of the program

This program performs the **spectral modelling** part of the imager processing. The spectral modelling aims to estimate 4 broadband radiances from the subsampled-calibrated imager radiance images (SEVIRI or MS7). These broadband radiances are:

- solar reflected unfiltered radiance $L_{sol} = \int L_{sol}(\lambda)\, d\lambda$,
- thermally emitted unfiltered radiance $L_{th} = \int L_{th}(\lambda)\, d\lambda$,
- shortwave GERB filtered radiance $L_{sw} = \int (L_{sol}(\lambda) + L_{th}(\lambda))\, \phi_{sw}(\lambda)\, d\lambda$,
- longwave GERB filtered radiance $L_{lw} = \int (L_{sol}(\lambda) + L_{th}(\lambda))\, \phi_{lw}(\lambda)\, d\lambda$,

More details about the spectral modelling part of the processing can be found in the RMIB document [6].

Notes

- The L_{sol} and L_{sw} images are only estimated when the SEVIRI visible channels are available.
- It is known that the different GERB detectors do not have exactly the same spectral response (SR) curves $\phi_{sw}(\lambda)$ and $\phi_{lw}(\lambda)$. For the spectral modelling, the used SR curves are the averaged SR curves for all the detectors as explained in [2].

Program Inputs / Output

inputs	outputs
configuration file	L_{sol} image
header file	L_{th} image
subsampled SEVIRI radiance images	L_{sw} image
spectral modelling coefficients files (6 files)	L_{lw} image
viewing and solar zenith angle images	

Operational program execution

The operational program has been executed on the 5 sets of input imager data described before. This execution is done using the commands:

- imager_spectralmod 20001851800 seviri_cfg.txt -v
- imager_spectralmod 20001851200 seviri_cfg.txt -v
- imager_spectralmod 20001852400 seviri_cfg.txt -v
- imager_spectralmod 20002051500 ms7_A_cfg.txt -v
- imager_spectralmod 20002061000 ms7_B_cfg.txt -v

3.2 Execution tests - Passed

Purpose of the test

Check the good execution of the program and the resources needed for the execution.

Tests methodology

The good execution of the various steps of the processing has been checked for the 5 test slots described before. During the execution, the informations displayed on the standard output (we used the verbose mode -v) are checked. It is also checked that all the expected output files have been generated. For the slot 2 (at midnight) it is tested that the solar and shortwave radiance images are not created.

Execution time

imaging device	total (s)	user (s)	system (s)
SEVIRI	2s	1s	0.7s
MS7	1s	0.5s	0.2s

The difference in computation time is due to the higher number of imager pixels for SEVIRI ($13\,10^6$) than for MS7 ($6\,10^6$).

3.3 Numerical tests - Passed

Object of the test

Validate the numerical values in the resulting images

Test methodology - criteria

The SEVIRI test slot 3 is used for this test. This test is done at the center of the Earth's disk. The numerical value at this location can be accessed using the **im_info** program. Test procedure:

1. the estimated spectral modelling values are evaluated at the centre of the Earth,
2. the viewing zenith angle and the solar zenith angle are evaluated at the centre of the Earth disk,
3. the regression parameters are evaluated for these angles by editing the spectral modelling parameters files,
4. the numerical evaluation of the 4 broadband radiances is done using a pocket calculator,
5. these values are finally compared with the values generated by the **imager_spectralmod** operational program that have been estimated in (1). These values must be the same.

3.4 Images visualisation tests - Passed

Object of the test

Validate the numerical values in the resulting images.

Tests methodology - criteria

The estimated unfiltered solar reflected and thermally emitted radiance images are visualised using the **viewrma** program. On these images it is checked that:

- for an optically thick cloud and the sun close to the zenith, the unfiltered solar reflected radiance should reach about $L_{sol} \sim 280\, Wm^{-2}sr^{-1}$.
- for a high level cloud (for example in the tropical region), the thermal radiance may reach value as small as $L_{th} \sim 30\, Wm^{-2}sr^{-1}$.

4 imager_cloudanalysis

4.1 Program Description

Object of the program

This program performs the **cloud analysis** part of the imager processing. This part of the processing aims to estimate the potential cloud cover in the imager's FOV. This estimation is mainly based on:

- the imager solar channels reflectance images,
- the infrared brightness temperature image (estimated for one channel),
- the clear sky reflectance images (which result from the weekly processing) .

The resulting images are:

- a cloud flag image: an internal flag used at RMIB
- the cloud fraction image: the fraction of cloudy pixels in a 3*3 pixel box,
- the cloud mean cover image: the mean value of C for the cloudy pixels in the box,
- the cloud mean optical thickness image: the mean value of τ for the cloudy pixels in the box,
- the cloud phase image: a floating point estimator in the range [0,1]. The value 0 corresponding to clouds with all particle in the liquid phase and 1 to clouds formed of ice crystals.

Notes:

- More details about the cloud analysis part of the processing can be find in the RMIB document [5],
- the cloud analysis part of the imager processing is mainly based on the imager solar channels and is then only executed when these channels are available.

Program Inputs / Output

inputs	outputs
configuration file	cloud flag image
header file	cloud fraction image
SEVIRI reflectance images	cloud mean C image
SEVIRI clear sky reflectance image	cloud mean τ image
vza, sza and ra images	cloud phase image
configuration data	

Operational program execution

The operational program has been executed on the 5 test slots described before. This execution is done using the commands:

- imager_cloudanalysis 20001851800 seviri_cfg.txt -v
- imager_cloudanalysis 20001851200 seviri_cfg.txt -v
- imager_cloudanalysis 20001852400 seviri_cfg.txt -v
- imager_cloudanalysis 20002051500 ms7_A_cfg.txt -v
- imager_cloudanalysis 20002061000 ms7_B_cfg.txt -v

4.2 Execution tests - Passed

Purpose of the test

Check the good execution of the program and the resources needed for the execution.

Tests methodology - criteria

The good execution of the various steps of the processing has been checked for the 5 sets of test data described before. During the execution, the informations displayed on the standard output (we used the verbose mode -v) are checked. It is also checked that all the expected output files have been generated. For the seviri +slot 2 (at midnight) it is tested that the cloud analysis is skipped.

Execution time

imaging device	total (s)	user (s)	system (s)
SEVIRI	22s	13s	2s
MS7	23s	17s	1s

4.3 Cloud flag image test - Passed

Purpose of the tests

Validate the content of the cloud flag image generated by the **imager_cloudanalysis** program. This test is done for the MS7-A and MS7-B tests slots.

Visualisation tests

The original visible band count images and the corresponding cloud flag images are visualised using the visualisation program **viewrma**. Then, it is checked that:

- the pixel labelled as **cloudy** in the cloud flag image correspond to cloudy pixels in the visible band images,
- the pixel labelled as **shadowed** in the cloud flag image correspond to shadowed pixels in the visible band images (mainly at the borders of high level cloud formation),
- the pixels labelled as **insufficient contrasted** correspond to highly reflective pixels in the clear sky visible band reflectance images (snow or ice surface, for example on the Greenland).

Ratio of cloudy pixels test

The ratio between the number of cloudy pixels and the sum of clear and cloudy pixels

$$f_{cloudy} = \frac{N_{cloudy}}{N_{cloudy} + N_{clear}}$$

can be evaluated using the utilitary program **im_histo** with the **-d** command line option. It is known that on global scale this ratio must be about 50%. For the MS7-A test slot, the ratio of cloudy pixels is $f_{cloudy} \sim 55\%$ when the cloud detection threshold (see after) is set to $\tau_d = 1.5$.

Sensitivity to the cloud detection threshold τ_d test

A pixel is labelled as cloudy in the cloud flag image if the estimated optical thickness at $\lambda = 0.55\mu$ for this pixel is higher than a given threshold τ_d that is set in the configuration file.

Various cloud flag images can be generated for different value of this threshold. The use of a threshold value of $\tau_d = 1.0$ must produce a cloud flag image with thin cloud labelled as cloudy. Using higher threshold values (for example $\tau_d = 1.5$ or $\tau_d = 2.0$), the number of cloudy labelled pixel must decrease and some thin clouds (mainly thin cirrus) may become classified as clear.

Sensitivity to the shadow detection threshold C_s test

A pixel is labelled as shadowed if its C value if below a given threshold C_s that is set in the configuration file. Usually a threshold value $C_s = -0.1$ is used. The modification of this threshold at $C_s = -0.05$ must increase the number of shadowed pixel at the borders of cloudy areas.

Sensitivity to the minimal contrast $\Delta\rho_{min}$ between clear sky and overcast reflectance test

A pixel is labelled as un-sufficiently contrasted if the difference between the overcast reflectance ρ_{oc} and the clear sky reflectance ρ_{cs} is smaller than a given threshold $\Delta\rho_{min}$ that is set in the configuration file. Usually a threshold value of $\Delta\rho_{min} = 0.2$ is used. When this value is increased (for example at $\Delta\rho_{min} = 0.4$) the highly reflective areas (snow, ice and deserts) must be labelled as insufficiently contrasted.

4.4 Cloud fraction image test - Passed

Purpose of the test

Validate the content of the cloud fraction image generated by the **imager_cloudanalysis** program. This test is done for the MS7-A and MS7-B tests slots.

Cloud fraction image visualisation test

The cloud flag image image and the corresponding cloud fraction images are visualised using the visualisation program **viewrma**. Then, it is checked that:

- for all pixel labelled as cloudy in the flag image the corresponding pixel in the cloud fraction image has non-null value,
- the clear (but also the un-sufficiently contrasted, shadowed and undefined) pixel in the flag image have all a null cloud fraction value.
- the cloud fraction is 0 in large clear areas and 1 in large cloudy area. The intermediate values must appear only for broken cloud areas and also at the border of the clouds.
- the ocean-land borders must not be visible in the cloud fraction image.

4.5 Cloud mean cover image test - Failure

Purpose of the test

Validate the content of the cloud C image generated by the **imager_cloudanalysis** program. This test is done for the MS7 tests slots.

Visualisation tests

The image of the cloud mean cover C is visualised using the **viewrma** program. On this image one checks that:

- for all pixels in clear sky areas, the C value is null,
- for optically thick clouds, the C value must be close to 1.0. **To date the C value for overcast scenes has usually values in the range $C = [0.8, 0.9]$ and seldom reaches the expected $C = 1$ value**.
- the ocean-land borders must not be visible on the C image. For this test, the dynamic range of the image visualisation must be reduce for example to the range $C = [0, 0.2]$.

Actions

Actually, we do not understand why the C values for optically thick clouds does not reach $C = 1$. This can be due to: (i) an over-estimation of the overcast reflectance ρ_{oc} (this estimation is done using RTM) or (ii) an under-estimation of the visible band radiance due to an incorrect imager calibration. This is under investigation.

4.6 Cloud optical thickness image test - Passed

Purpose of the test

Validate the content of the cloud τ image generated by the **imager_cloudanalysis** program. This test is done for the MS7-A and MS7-B tests slots.

Visualisation tests

The image of the cloud mean cloud optical thickness τ is visualised using the **viewrma** program. On this image one checks that:

- for all pixels in clear sky areas, the τ value is null,
- the τ value may never be small than the threshold τ_d used to detect cloudy pixels,
- for optically thick clouds, the τ value must reach value higher than $\tau = 50$ (the maximum τ value for the processing is 400).
- the ocean-land borders must not be visible on the τ image,

4.7 Cloud phase image test - Passed

Purpose of the test

Validate the content of the cloud phase image generated by the **imager_cloudanalysis** program. This test is done for the Meteosat-7 tests slots.

Visualisation tests

The cloud phase images and the corresponding brightness temperature images are visualised using the visualisation program **viewrma**. Then, it is checked that:

- the cloud phase varies in the range [0,1],
- for clear areas the cloud phase is set to 0 (water cloud),
- the correspondence between brightness temperature and phase is correct (numerical test),
- the ice phase cloud (value 1) are mainly detected for depth convective clouds in the ITCZ,
- the marine cumulus on the Atlantic ocean are mainly featured as water clouds.

Sensitivity to the cloud phase estimation parameters tests

The cloud phase is estimated from the averaged brightness temperature for the cloudy pixels using a linear limited function. There are two parameters is the configuration files: $T_{B,ice}$ and $T_{B,water}$. Typical values for these parameters are:

$$T_{B,ice} = 245\,K$$
$$T_{B,water} = 265\,K$$

During this tests, it is checked that using low threshold values ($T_{B,ice} = T_{B,water} = 200\,K$) all cloudy pixels have cloud phase at 0 (water cloud). On the other hand, using high temperature threshold ($T_{B,ice} = T_{B,water} = 320\,K$) all cloudy pixels must be labelled as ice phase cloud.

5 imager_rad2flux

5.1 Program Description

Object of the program

This program performs the estimation of the solar and thermal fluxes from the unfiltered solar and thermal radiances. This estimation is done in two steps:

- first the best-suited angular models are selected for each pixel in the FOV,
- then the conversion from radiance to flux is done by application of these angular models.

Notes:

- More details about the angular conversion part of the processing can be found in the RMIB document [4],
- the estimation of the solar flux image is only done when the solar radiance image is estimated. That means when the imager solar channels are available,
- to date, for solar reflected radiation, the set of ADM from ERBE is used.

Program Inputs / Output

inputs	**outputs**
configuration file	solar ADM label image
header file	thermal ADM label image
cloud analysis results (=> **imager_cloudanalysis**)	solar flux image
surface identification results (=>**weekly processing**)	thermal flux image
vza, sza and ra images (=>**imager_common_proc**)	
set of ADM (ERBE, CERES)	
solar and thermal radiance images (=>**imager_spectralmod**)	

Operational program execution

The operational program has been executed on the 3 set of input SEVIRI data described before. This execution is done using the commands:

- imager_rad2fluxes 20001851800 seviri_cfg.txt -v
- imager_rad2fluxes 20001851200 seviri_cfg.txt -v
- imager_rad2fluxes 20001852400 seviri_cfg.txt -v
- imager_rad2fluxes 20002051500 ms7_A_cfg.txt -v
- imager_rad2fluxes 20002061000 ms7_B_cfg.txt -v

5.2 Execution tests - Passed

Purpose of the test

Check the good execution of the program and the resources needed for the execution.

Tests methodology - criteria

The good execution of the various steps of the processing has been checked for the 5 sets of test slots described before. During the execution, the informations displayed on the standard output (we used the verbose mode **-v**) are checked. It is also checked that all the expected output files have been generated. For the slot 2 (at midnight) it is tested that the solar flux image is not created.

Execution time

imaging device	total (s)	user (s)	system (s)
SEVIRI	2.1s	1.6s	0.4s
MS7	2.8s	0.7s	0.2s

label	model
0	averaged ERBE model
1	ERBE model 1 (clear ocean)
2	ERBE model 2 (clear land)
3	ERBE model 3 (clear snow)
4	ERBE model 4 (clear desert)
...	...
12	ERBE model 12 (overcast)
...	not used
254	lambertian model
255	no flux model (result flux = 0)

Table 2: Code used for the solar ADM label image that serves for the conversion of the solar radiance into solar flux. This table correspond to the use of the ERBE models. In the future, the code will be modify to use the CERES models (about 180 models).

5.3 Solar ADM label image tests - Passed

Purpose of the tests

Validate the numerical values in the image of the best-suited angular model to use for the conversion of solar radiance into solar flux. The code used for this image is given in table (3).

Visualisation tests

For this test the solar ADM label images are simply visualised for the 2 Meteosat test slots. These images can be visualised in false colour using the **viewrma** program with the appropriate colormap (add **-c 3** on the command line). On this image it is checked that:

- for clear sky areas, the selected ADM models for solar radiation correspond to the surface types (see the code in table 3 hereafter),
- for overcast areas the values in the solar ADM label image is 12 (these areas appear in white in the false colour image),
- the partly and mostly cloudy values are encountered at the borders of the cloud formations and in broken cloud areas.

Image of surface type visualisation test

The surface type image are visualised using the visualisation program **viewrma** with the appropriate colormap (command line option **-c 3**). The local noon clear sky reflectance image is also visualised[1]. Then, it is checked that:

- the kind of surface is coherent with the image of visible band clear sky reflectance at local noon,

[1] These images are generated by the weekly processing and are stored in the directory planet:/gerbdata/nic/ImagerProcess/Surface.

label	surface type
0	no surface identification
1	ocean
2	land
3	snow and ice
4	desert
5	mix ocean-land

Table 3: Code used for the surface type image generated by the weekly processing. In the future, it is possible that this code will be enlarged by some additional classes from the IBGP.

- the mix ocean-land pixels are located at the borders of the oceans, lakes or rivers,
- the “no surface identification” label is only used for pixel out of the Earth disk.

Notes :

- the surface type image is not generated by the operational imager processing but by the **weekly processing**. The “Image of surface type visualisation test” aims just to validate the content of this image before the **imager_rad2fluxes** program.
- the code used for the surface type image is given in table (3).

Special classes tests

This test aims to validate the used of the “special” label in the solar ADM label image. For this test, the solar ADM label image is visualised using the viewrma program with the appropriate colormap (-c 3), then it is checked that:

- the ’no flux model’ (label 255) appears (only) for the imager pixels out of the Earth disk,
- the ’lambertian model’ (label 254) appears for the pixels with solar zenith angle higher than 90^o and for the pixels labelled as shadowed during the cloud analysis process,
- the ’ERBE averaged model’ (label 0) appears for pixels with solar zenith angle in the range $[80^0, 90^o]$.

5.4 Solar flux image tests - Passed

Purpose of the tests

Validate the numerical values in the image of the solar flux which is generated by the operational program **imager_rad2fluxes**..

Visualisation tests

For this test the solar flux images are simply visualised for theMS7-A and MS7-B test slots. These images can be visualised in grey scale using the **viewrma** program. On this image it is checked that:

- no discontinuities (artefacts) are introduced in this image by the various steps of the processing,
- the solar flux value is never negative,
- for a opticallly thick cloud area close to the sun sub-point, the solar reflected flux must reach value of about $F_{sol} \sim 800Wm^{-2}$,
- the flux image value looks like the original visible band narrow-band radiance image.

Comparison with the development program 'apply_adm_sol'

For this test the solar flux image is generated using the development program **apply_adm_sol**. This program takes as input

- the solar radiance image (generated by the imager_spectralmod program),
- the image of solar ADM label (generated by the imager_rad2fluxes program),
- the listing of the ADM files (see configuration files),
- the 3 images of viewing zenith, solar zenith and relative azimuth angles (generated by the imager_common_proc).

The resulting solar flux images must be the same as the ones generated by the operational program (same sizes and same data field values). The program **imdiff** is used to compare these two images.

Used development program

- **apply_adm_sol** : perform the angular conversion for the solar radiation,
- **imdiff** : compare two images.

5.5 Thermal flux image tests - Passed

Note:

In the current implementation of the ground segment, a single ADM is used for the angular modelling of thermal radiation. So, for all pixel in the imager FOV the same ADM label is used (label 0). For this reason, contrary to the solar ADM label image, the thermal ADM label image is not generated by the **imager_rad2fluxes** program.

Purpose of the tests

Validate the numerical values in the image of the thermal flux which in produced by the **imager_rad2fluxes** oparational program..

Visualisation tests

For this test the thermal flux images are simply visualised for the MS7-A and MS7-B test slots. These images can be visualised in grey scale using the **viewrma** program. On this image it is checked that:

- no discontinuities (artefacts) are introduced in this image by the various steps of the processing,
- the thermal flux value is never negative,
- for hot surface area (desert) the theraml flux by reach value as high as $F_{th} \sim 400Wm^{-2}$.
- for cold high level clouds, the thermal flux may be as weak as $F_{th} \sim 100Wm^{-2}$.
- the flux image value looks like the original infrared narrow-band radiance image of Meteosat-7.

Comparison with the development program 'apply_adm_th'

For this test the thermal flux image are generated using the development program **apply_adm_th**. This program takes as input

- the thermal radiance image (generated by the imager_spectralmod program),
- the ADM used for the thermal radiation angular conversion (see configuration files),
- the image of viewing zenith angle (generated by the imager_common_proc).

The resulting thermal flux images must be the same as the ones generated by the operational program (same sizes and same data field values). The program **imdiff** is used to compare these two images.

Used development program

- **apply_adm_th** : is used to perform the angular conversion for the solar radiation,,
- **imdiff** : is used to compare two images.

6 Routines from libbats.a

6.1 Introduction

The programs described in this document make intensive calls to some important routines from the BATS library. The test methodology for these routines are described in this section. Some program have been writen to test these routines. They are located in the directory planet:/usr/people/nic/ImagerProcess/Test_V3/TestLibRoutine/

6.2 GP_SlotToUnixTime - Passed

Object of the routine

Compute the time for a character's string that complies with one of the 2 formats used at RMIB for the GERB ground segment:

formats	prefix syntax
JULIAN_TIME_FORMAT	YYYYDDDHHMM
STD_TIME_FORMAT	YYYYMMDDHHMMSS

The returned time is expressed in second in the UNIX epoch (starting the January 1st 1970 at 0h 00' 00"). The input character's string correspond always to UTC time.

Syntax

double GP_SlotToUnixTime (char *prefix);

Test objective

- be sure the character's string s are correctly interpreted,
- be sure the conversion from string to time is done for UTC time.

Test methodology

The utilitary program **meteosat_time** is used to check this routine.

The following calls must return '60':

meteosat_time 19700010001

meteosat_time 19700101000100

The following calls must the error value (-1.0):

meteosat_time 1970001

meteosat_time 1970010100010090

meteosat_time 19680011200

meteosat_time blabla

These tests have been done for various RMIB computer with different Operating Systems (Linux, IRIX, OSF, ...).

6.3 GP_TimeExtractHour - Passed

Object of the routine

Extract the part of the character's string str that corresponds to the hour and minute.

Syntax

char *GP_TimeExtractHour(char *str);

Test methodology

The utilitary program **prefix2slot** is used to check this routine.

The following calls should return '1200':

prefix2slot 20000011200

prefix2slot 20000101120000

The following calls must the error value (-1.0):

prefix2slot 1970001

prefix2slot 19700101000010090

prefix2slot 19680011200

prefix2slot blabla

These tests have been done for various RMIB computer with different Operating Systems (Linux, IRIX, OSF, ...).

6.4 GP_EarthSunDistance - Passed

Object of the routine

Compute the distance between the Earth and the Sun in astronomical unit.

Syntax

double GP_EarthSunDistance(time_t t);

Test methodology

The Earth to Sun distances are computed for each day of the year 2000 at 12 UTC using the test program **test_earthsundist.c**. This distance is stored in a file which can be visualised using **gnuplot** (see figure 1). On this graph it is checked that:

- this distance is minimal the 2 January of each year,
- the variation of the Earth-sun distance is a bit more than 3% during the seasons.

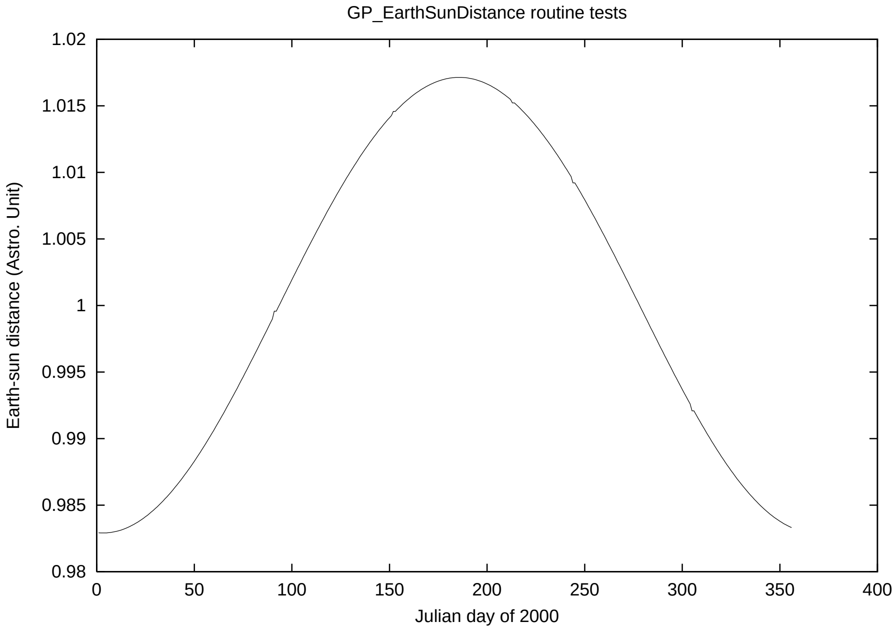

Figure 1: Plot for the **GP_EarthSunDistance routine tests.**

6.5 CalculateViewingAngles - Test methodology to be defined

Object of the routine

Compute the viewing zenith, solar zenith and relative azimuth angles for a points on the Earth's surface specified by its (lat, lon) geocentric coordinates and for a sun position in the sky specified by its (lat, lon) coordinates.

Syntax

ADM_angle CalculateViewingAngles(double lat_earth,double lon_earth,double lat_sun,double lon_sun);

Test methodology

To be defined.

6.6 ADM_SceneInterpolate - Passed

Object of the routine

Interpolation within a ADM_scene structure to get the BRDF (Bidirectional Reflectance Distribution Function) and the aledo values.

Syntax

double ADM_SceneInterpolate(ADM_scene *adm, ADM_angle geom, double *albedo);

Test using constant model

A constant ADM model files has been generated using the utilitary program **adm_create_const**. This file is filled with 1.0 value for the BRDF and the albedo at all angle. Then it must be checked that the interpolatuion at any angle within this model gives the 1.0 values for the BRDF and the albedo. The test program **testadm** is used.

Test of the interpolation process

The constant ADM file generated for the previous test is here edited and modified (some values are set to 0). It is checked that the trilinear interpolation is done correctly. The test program **testadm** is used.

6.7 C2tau- Passed

Object of the routine

Estimation of the cloud optical thickness from the C measured value, the angle of view and the parameters stored as 4 ADM_scene structures. The used formula is :

$$\tau = \tau_0 \, (\frac{C}{A - B\,C})^{\theta}$$

where the 4 parameters $\{\tau_0, \theta, A, B\}$ are dependent on the viewing geometry:

$$\begin{aligned} \tau_0 &= \tau_0(\theta_v, \theta_s, \phi) \\ \theta &= \theta(\theta_v, \theta_s, \phi) \\ A &= A(\theta_v, \theta_s, \phi) \\ B &= B(\theta_v, \theta_s, \phi) \end{aligned}$$

Syntax

double C2tau(double C, ADM_angle *geom, ADM_scene *tau0, ADM_scene *chi, ADM_scene *A, ADM_scene *B);

Test using constant model

4 constant ADM model files has been generated using the utilitary program **adm_create_const**. These files are filled with the “BRDF” values given hereafter and null albedo.

$$\begin{aligned} \tau_0 &= 11.073500 \\ \theta &= 0.856808 \\ A &= 1.00527 \\ B &= 1.00046 \end{aligned}$$

The test program **testC2tau**.c is used to build the τ versus C curve (see figure 2).

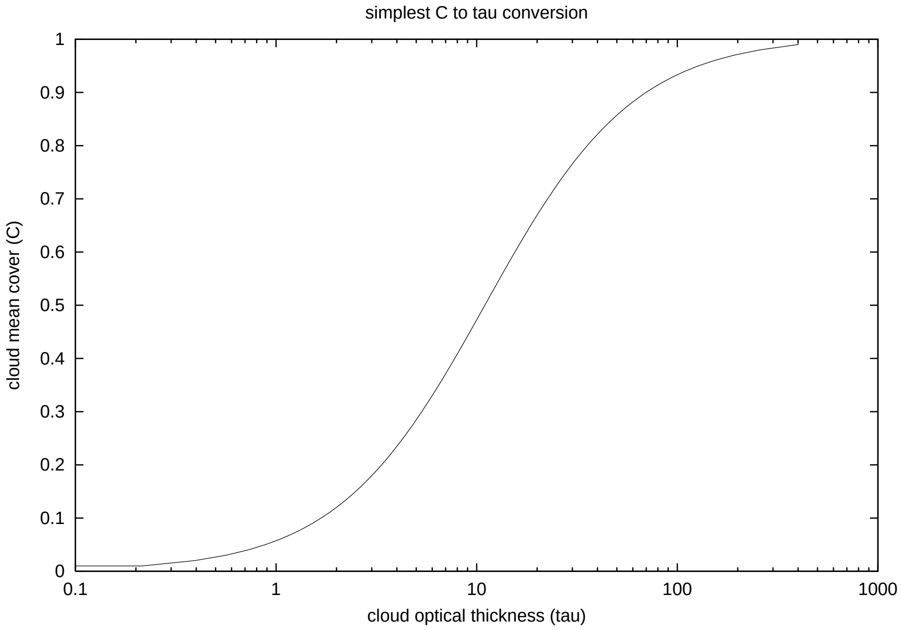

Figure 2: C to τ conversion formula.

References

[1] Clerbaux N. Derivation of toa fluxes from meteosat-7 imager. Internal Document MSG-RMIB-GE-TN-0030, RMIB, November 1999.

[2] Clerbaux N. Correction of the dispersion in the gerb's detector spectral response curves. Internal document, RMIB, May 2000.

[3] Clerbaux N. Provisional calibration coefficient for seviri. Internal document, RMIB, May 2000.

[4] Clerbaux N. and Dewitte S. Rmib gerb processing - seviri processing : Angular dependency models. Reference Document MSG-RMIB-GE-TN-0009, RMIB, August 1999.

[5] Clerbaux N. and Dewitte S. Rmib gerb processing - seviri processing : Scene identification. Reference Document MSG-RMIB-GE-TN-0008, RMIB, August 1999.

[6] Clerbaux N. and Dewitte S. Rmib gerb processing - seviri processing : Spectral modeling. Reference Document MSG-RMIB-GE-TN-0004, RMIB, August 1999.