



# Projection of Meteosat images into World Geodetic system WGS-84 matching Spot/Vegetation grid

Bruno COMBAL, Josué NOEL

EUR 23945 EN - 2009

The mission of the JRC-IES is to provide scientific-technical support to the European Union's policies for the protection and sustainable development of the European and global environment.

European Commission  
Joint Research Centre  
Institute for Environment and Sustainability

**Contact information**

Address: Global Environment Monitoring Unit, Institute for Environment and Sustainability, Joint Research Centre, via E. Fermi, 2749, I-21027 Ispra (VA), Italy  
E-mail: [bruno.combal@jrc.ec.europa.eu](mailto:bruno.combal@jrc.ec.europa.eu)  
Tel.: +39 03 32 78 93 78  
Fax: +39 03 32 78 90 73

<http://ies.jrc.ec.europa.eu/>  
<http://www.jrc.ec.europa.eu/>

**Legal Notice**

Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use which might be made of this publication.

***Europe Direct is a service to help you find answers  
to your questions about the European Union***

**Freephone number (\*):  
00 800 6 7 8 9 10 11**

(\* ) Certain mobile telephone operators do not allow access to 00 800 numbers or these calls may be billed.

A great deal of additional information on the European Union is available on the Internet. It can be accessed through the Europa server <http://europa.eu/>

JRC 52438

EUR 23945 EN  
ISBN 978-92-79-12953-7  
ISSN 1018-5593  
DOI 10.2788/2452

Luxembourg: Office for Official Publications of the European Communities

© European Communities, 2009

Reproduction is authorised provided the source is acknowledged

*Printed in Italy*

## Table of contents

<b>1</b>	Introduction.....	5
<b>2</b>	Systems descriptions .....	6
2.1	METEOSAT Second Generation.....	6
2.2	Spot VEGETATION.....	11
2.2.1	Satellite description .....	11
2.2.2	Geometrical processing .....	12
2.2.3	NDVI synthesis .....	14
2.3	Projection of MSG images into Spot/VEGETATION grid.....	14
<b>3</b>	Quality analysis of the re-projection .....	18
3.1	Consistency of projected and resampled MSG images with SPOT/VEGETATION .....	18
3.2	IFOV distortion .....	22
3.3	METEOSAT band inter-registration and inter-correlation analysis.....	23
<b>4</b>	Conclusion.....	26
<b>5</b>	Annex.....	27
5.1	Mosaicing code .....	27
5.2	Projection code.....	31
<b>6</b>	References .....	36

## List of tables

Table 1. SEVIRI bands (Meteosat 8).....	6
Table 2. Spot/VEGETATION bands .....	12
Table 3. Characteristics of the 4 LandSAF geographical areas. Each region is defined by the corner positions relative to an image of 3712 columns per 3712 lines with indices starting at 1 in the North and West [SAF/LAND/MF/PUM_AL/1.4] .....	15
Figure 4. Mosaic of NAfr and SAfr regions done with mosaic_msg.py .....	15

## List of figures

Figure 1. MSG/SEVIRI, Channel 3.....	7
Figure 2. Channel 12, high resolution: only two split windows are used.....	7
Figure 3. High resolution and standard resolution. The lines reduced or lines + columns reduced dissemination formats will only be used in exceptional cases....	8
Figure 4. Meteosat Earth imaging principle.....	9
Figure 5. Alignment and numbering of the non-HRV pixels .....	9
Figure 6. MSG Level 1.5 ground resolution map (N-S direction), after [EUM/MSG/ICD/105] .....	10
Figure 7. MSG Level 1.5 ground resolution map (E-W), after [EUM/MSG/ICD/105] .....	10
Figure 8. geometric distortion of VEGETATION due to Earth curvature .....	12
Figure 9. The LandSAF products are split on 4 windows: Europe (Euro), Northern Africa (NAfr), Southern Africa (SAfr), South America (SAme) .....	14
Figure 10. Geographical coordinates .....	16
Figure 11. Coordinate frames for Geostationary Satellite Projection .....	17
Figure 12. Projected MSG image and Spot/Vegetation (red bands). .....	18
Figure 13. The correlation is analysed in windows of 40 pixels width (red box)..	19
Figure 14. Areas with a spatial correlations better than 0.7 are good enough to estimate the images shifts.....	20
Figure 15. x and y offsets per sub-windows between Spot/Vegetation and Meteosat images.....	20
Figure 16. Spot/VEGETATION has a 10km buffer along the coast. ....	21
Figure 17. Sub-windows x- shifts, where correlation is better than 0.7.....	21
Figure 18. Sub-windows y-shifts, where correlation is better than 0.7.....	22
Figure 19. Sub-windows shift (shift norm), where correlation is better than 0.7	22
Figure 20. Effect of pixel IFOV on MSG projected image.....	23
Figure 21. Correlations between bands C1 and C2 .....	24
Figure 22. Correlation between bands C1 and C3.....	24
Figure 23. Windows shift (misregistration norm) between channels C1 and C2, where correlation is better than 0.7 .....	25
Figure 24. Windows shift (misregistration norm) between channels C1 and C3, where correlation is better than 0.7 .....	25

# 1

## Introduction

DG-JRC (IES, MONDE action) worked on the development of an integrated system (named e-station, where 'e' stands for environment) able to ingest data broadcasted in the EumetCast flux, with the goal of generating environmental indicators as soon as data are received for addressing a list of pre-identified environmental issues.

The first sets of indicators were build on Spot/VEGETATION derived factors (namely VGT4africa product portfolio), and MSG/SEVIRI derived factors (such as LandSAF products).

The two Earth observing system are greatly different: Spot/VEGETATION is a polar orbiting system which offers an almost constant cross segment geometric quality, while MSG is a geostationary satellite which offers a very high rate of observation (up to 1 image every 15 minutes) but maximises images distortion.

In order to offer the possibility to seamlessly access to the whole range of information provided by those two types of Earth Observing system, the MSG derived images are re-projected to the Spot/VEGETATION projection and grid: plate carrée, WGS84, reference in pixel centre, square pixel size equals  $(1/112)^\circ$ , round latitude and longitude falling in a pixel centre. Among all the possible application, one of the first is to combine MSG factors related to its spectral channels (such as surface temperature) and Spot/Vegetation higher resolution for observing land surface.

This document reports for the Spot/VEGETATION and MSG/SEVIRI system characteristics, the re-projection method to transform MSG geostationary satellite projection view to Spot/VEGETATION projection system and grid. The distortion introduced by the geostationary observing system is also estimated.

# 2 Systems descriptions

## 2.1 METEOSAT Second Generation

Note: most information in this section comes from <http://www.eumetsat.int>

EUMETSAT, the European Organisation for the Exploitation of Meteorological Satellites, operates Meteosat satellites since 1977. In 2002 Meteosat Second Generation (MSG) satellite, Meteosat-8, brought significant improvements through a 12 spectral bands (instead of 3 in the first generation) of the Spinning Enhanced Visible and Infrared Imager (SEVIRI).

SEVIRI delivers daylight images of the weather patterns with a resolution of 3 km, plus atmospheric pseudo-sounding and thermal information. The High Resolution Visible (HRV) channel has a resolution of 1 km. The full disc view allows frequent sampling, every 15 minutes, enabling monitoring of rapidly evolving events.

In addition to the SEVIRI, the MSG satellites carry the Geostationary Earth Radiation Budget (GERB) instrument, which provides data on reflected solar radiation and thermal radiation emitted by the Earth and its atmosphere.

The High rate SEVIRI images are acquired every 15 minutes for each 12 spectral channels. The data consist of geographical arrays of various sizes of image pixels, each pixel being encoded on 10 bits representing the received radiation from the Earth and its atmosphere. For MSG1 (Meteosat 8), the spectral bands are (EUM/OPS/TEN/05/2561):

Band	Name	Nominal central wavelength	Nominal spectral band ( $\mu\text{m}$ )
1	Visible band	0.635 $\mu\text{m}$	0.56 – 0.71
2	Visible band	0.81 $\mu\text{m}$	0.74 – 0.88
3	Near Infrared (discriminate ice/water clouds)	1.64 $\mu\text{m}$	1.50 – 1.78
4	Infrared band (cloud and fog detection)	3.92 $\mu\text{m}$	3.48 – 4.36
5	Water vapour	6.25 $\mu\text{m}$	5.35 - 7.15
6	Water vapour	7.35 $\mu\text{m}$	6.85 – 7.85
7	Infrared (discriminate ice/water clouds)	8.70 $\mu\text{m}$	8.30 – 9.10
8	Ozone	9.66 $\mu\text{m}$	9.38 – 9.94
9	Infrared	10.8 $\mu\text{m}$	9.80 – 11.80
10	Infrared	12.0 $\mu\text{m}$	11.00 – 13.00
11	Carbon dioxide	13.40 $\mu\text{m}$	12.40 – 14.40
12	Broadband high resolution visible		0.5 – 0.9

Table 1. SEVIRI bands (Meteosat 8).

The bands spectral responses are given in document Meteosat-8/MSG 1 - Spectral Response (EUM/OPS/TEN/05/2562).

The 11 first channels have an image size of 3712x3712 pixels (e.g. Figure 1), while channel 11 has an image size of 5568x11136 pixels as illustrated on Figure 2 (EUM/MSG/SPE/057, p. 27).

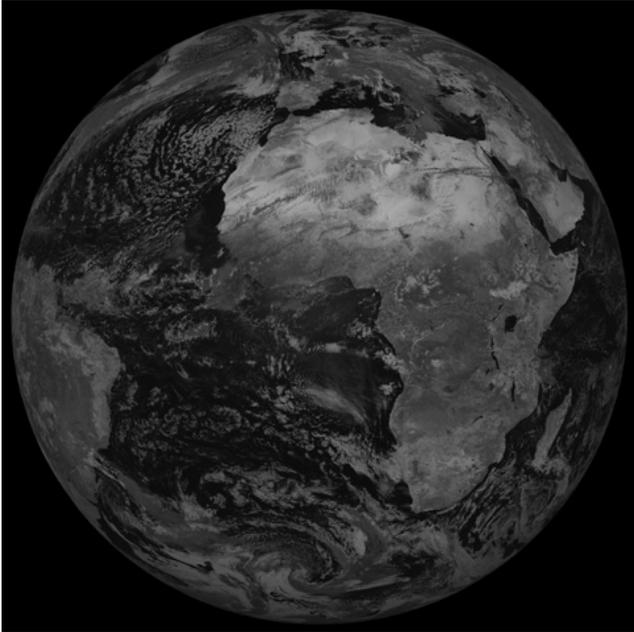


Figure 1. MSG/SEVIRI, Channel 3.



Figure 2. Channel 12, high resolution: only two split windows are used

The High Resolution Visible mode consists in split windows where the resolution is higher. From 31 August 2005 onwards, the lower window shifts to follow the daily illumination. The configuration is as follows:

- Between 00:00 and 14:00 UTC the lower HRV window is east-justified so as to optimise the coverage of the Indian Ocean region;
- Between 14:00 and 17:00 UTC the lower HRV window moves every hour, on the hour incrementally westwards, until it becomes west-justified;
- Between 17:00 and 00:00 UTC the lower HRV window is west-justified to optimise the coverage of the Atlantic Ocean.

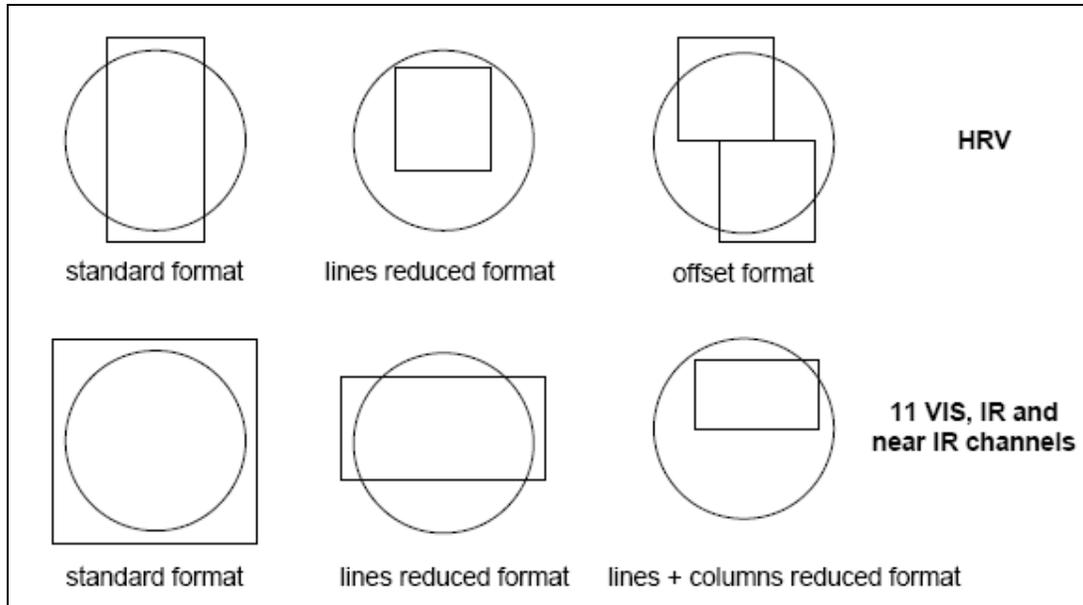


Figure 3. High resolution and standard resolution. The lines reduced or lines + columns reduced dissemination formats will only be used in exceptional cases.

MSG dissemination has two channels, a High Rate Information Transmission channel (HRIT) and a Low Rate Information Channel (LRIT).

The HRIT channel aims at Primary Data Users who require receiving and processing a comprehensive set of information, like European meteorological centres, regional area forecast centres, research laboratories. The LRIT channel will only contain sub-sets of data or data sets compressed by higher factors. These will usually satisfy Secondary Data Users with limited capability in terms of acquisition and processing like smaller National Meteorological Services, universities, commercial companies, individuals etc.

The scanning of the Earth disk is obtained by using the satellite spin (100 rpm +/- 1%) in the East-West direction and by stepping a flat scan mirror in the South-North direction after each East-West line, to set up the instrument for acquiring the scan of image data (Figure 4). One complete revolution of the satellite lasts 0.6 seconds of which only about 30 ms are available over the Earth disk to acquire one scan. After the 30ms spent imaging the Earth, the remaining 570ms are used mainly for scan mirror stepping, data transmission and deep space data acquisition for Direct Current Removal (DCR). The image nominal repeat cycle is 15 minutes, including on-board radiometric calibration and scan mirror retrace. Shorter repeat cycles are programmable if an image of a reduced area of Earth is required.

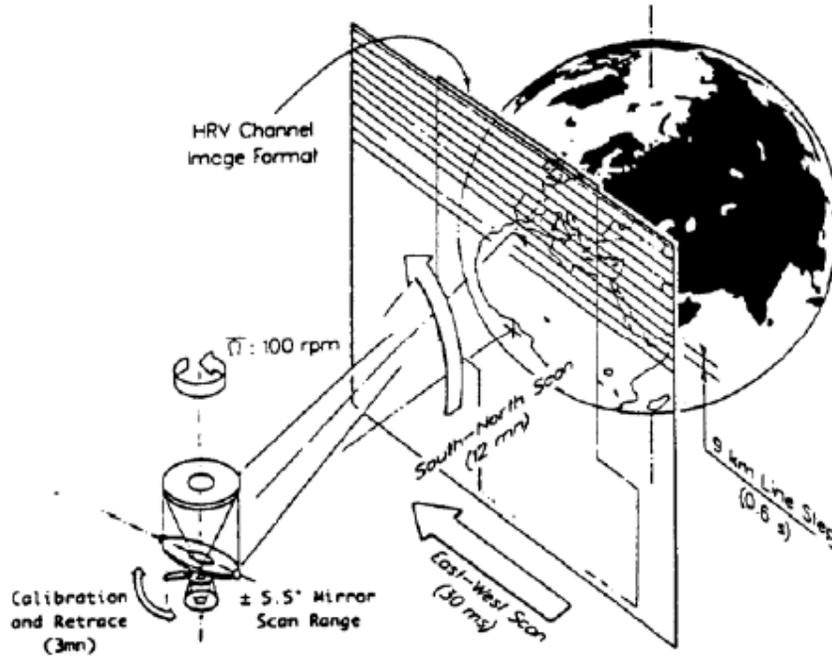


Figure 4. Meteosat Earth imaging principle

For the non-HRV images, the nominal geostationary projection centre ( $0^\circ$  longitude,  $0^\circ$  latitude) coincides with the middle of the pixel that has the line and column number (1856,1856), where the pixel numbering starts in the South-Eastern corner of the image with line and column number (1,1), see Figure 5 [EUM/MSG/ICD/105].

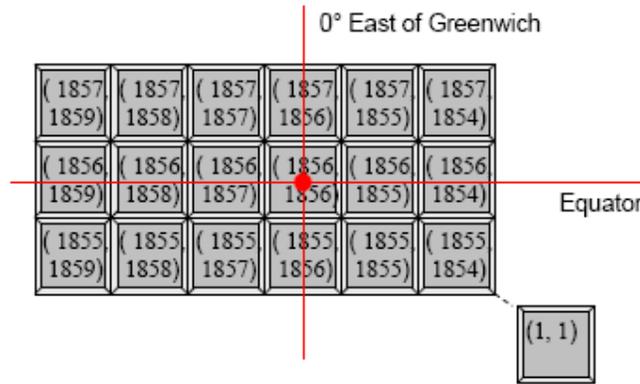


Figure 5. Alignment and numbering of the non-HRV pixels

The ground resolution varies with latitude and longitude, which is a direct consequence of the image acquisition process. On Figure 6 and Figure 7, the ground pixel size varies from 3.1km (dark color), 4km, 5km, 6km, 8km, and 11km (darker to lighter grey). Over Africa, the pixel size is of about 4km in X and Y directions (5km in X-direction on Eastern Africa).

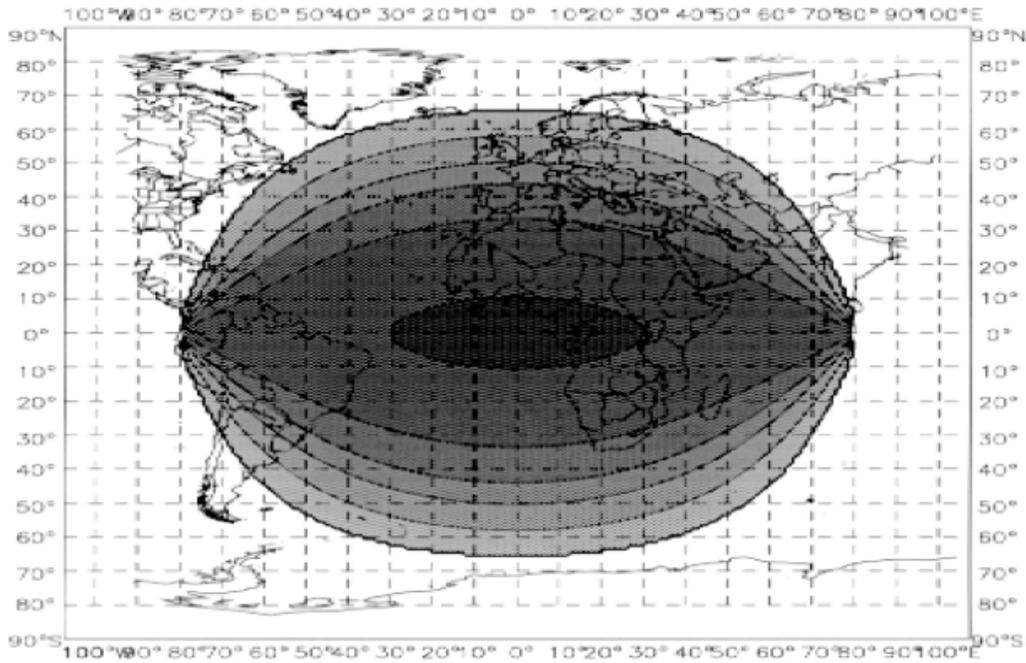


Figure 6. MSG Level 1.5 ground resolution map (N-S direction), after [EUM/MSG/ICD/105]

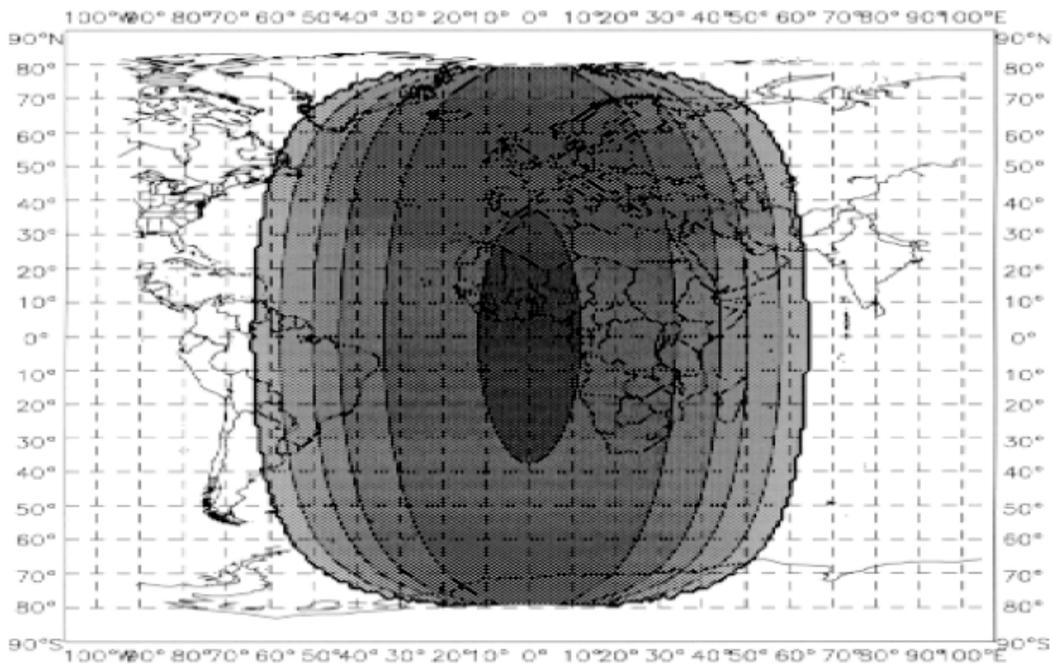


Figure 7. MSG Level 1.5 ground resolution map (E-W), after [EUM/MSG/ICD/105]

The geometric accuracy of the images after ground processing is formulated in a statistical sense, either over 1 day (absolute error, error over 500 lines, error over 6 lines) or in a relative sense between two consecutive images.

The absolute error within one image is about 1.32km in the N-S direction and 0.15km in the E-W direction [EUM/MSG/ICD/105].

The phenomenon of misregistration is mainly due to the thermo-elastic variations within the SEVIRI (both between the cold & warm focal planes and to a lesser extent between channels within a given focal plane). The misregistrations characterised on ground are of limited value after the launch, due to

its thermal and mechanical influences. Therefore, the on-ground processing is an auto-adaptive method that continuously re-estimates and corrects the focal plane misregistration based on the newest observations taken from the image. It has a typical integration time of one day, but shorter periods are considered following eclipse or manoeuvre situations.

The inter-channel registration is of 0.45km (Table 2) for any combination of channels [EUM/MSG/ICD/105], which is the sum of errors arising from interpolation methods used (0.27km) and errors in the SEVIRI focal plane model (0.18km). The grid of HRV images is such that 3x3 pixels of the HRV image are registered to 1 pixel of the VIS/IR channels.

More up-to date geometric quality and inter-channel registration are found in the header record of the MSG images files [EUM/MSG/ICD/105, p. 39].

<b>Channel Groups</b>	<b>Residual Misregistration Requirement (km)</b>	<b>Residual Misregistration Budget (km)*</b>
<b>VNIR</b>	0.6	0.45
<b>Warm (HRV +VNIR)</b>	0.6	0.45
<b>Window (IR3.9/IR8.7/IR10.2/IR12.0)</b>	0.6	0.45
<b>Cold (All except HRV and VNIR)</b>	0.75	0.45
<b>ALL</b>	1.5	0.45

Table 2. Interchannel registration, distances at sub-satellite point (SSP).

## 2.2 Spot VEGETATION

### 2.2.1 Satellite description

The VEGETATION Programme cooperation agreement was signed on May 25th, 1994 by France, the European Commission, Belgium, Italy and Sweden to develop and operate the VEGETATION instruments. The first satellite component (VEGETATION 1) of the programme was launched on 24th March 1998 onboard the SPOT 4 satellite, while the second instrument was launched onboard SPOT 5 on 4th May 2002.

The VEGETATION instruments deliver measurements specifically tailored to monitor land surface properties with a frequency of about once a day on a global basis and at a spatial resolution of 1 km (in fact 5 times over 6 days at the equator, once a day at 30° latitude, and twice a day at latitudes higher than 60°).

The wide angle (101°) optical imaging instrument is the core of the VEGETATION payload. As SPOT satellites average altitude is 830 km the ground swath width is 2200 km. The CCD linear arrays of the instrument collect radiation reflected by the earth's surface in 4 spectral bands (Table 2) to convert it into usable electrical signals.

Blue (B0)	0.430 $\mu\text{m}$ – 0.470 $\mu\text{m}$
Red (B2)	0.610 $\mu\text{m}$ – 0.680 $\mu\text{m}$
Near infrared	0.780 $\mu\text{m}$ – 0.890 $\mu\text{m}$
Short wave infrared	1.580 $\mu\text{m}$ – 1.750 $\mu\text{m}$

Table 2. Spot/VEGETATION bands

The CCD technology avoids the panoramic distortion of scanning devices and limits geometrical distortion of images to the effect of Earth curvature (Figure 8). Each spectral band has its own telecentric optics with the associated detector unit in the focal plane. A highly stable structure ensures the alignment of all 4 cameras optical axes to within 5 arc second.

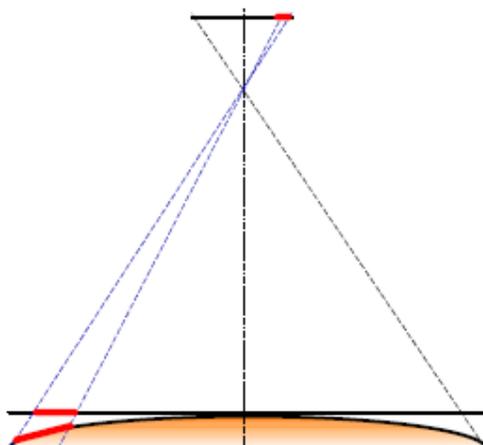


Figure 8. geometric distortion of VEGETATION due to Earth curvature

## 2.2.2 Geometrical processing

The main objective of geometrical correction is to compute the location in the raw image of each pixel of a map-projected product. The most important error elements are described hereafter, with their order of magnitude and the correction applied, when possible.

### 2.2.2.1 Dating

There is an uncertainty of about 10 milliseconds in the on-board time, coming from the on-board time data accuracy (4ms) and the drift of the on-board clock. This leads to a satellite location error of less than 0.1 km.

### 2.2.2.2 Satellite location

The satellite location is estimated using two methods: the first one uses an orbit model, called MADRAS, which gives a location accuracy of about 0.3 km; the other one is based on measures of an accurate onboard navigator called DORIS/DIODE. This gives a much better location accuracy of about 10 meters. This second method is normally, but, in case of DORIS/DIODE service interruption, MADRAS model replaces DORIS/DIODE.

### 2.2.2.3 Satellite absolute attitude

The SPOT satellite attitude movements are measured by a set of gyrometers, but the absolute attitude is known only from a sun sensor and an earth sensor, which does not give accuracy better than 0.6 milliradians (peaks around 1 milliradian). This leads to a 0.5 km RMS location error in the images. This quite big error needs to be corrected to reach VEGETATION mission requirements: a Ground Control Point workshop was used for that purpose for the VEGETATION 1 instrument onboard SPOT 4,

whereas for VEGETATION 2 more accurate measurements provided by a star tracker mounted on-board the SPOT 5 platform allows to reach a 50 m accuracy without the use of ground control points. The star tracker uses stars measured position to estimate the satellite's attitude with accuracy better than  $0.01^\circ$ .

#### 2.2.2.4 Satellite relative attitude

The gyrometers data are used to measure the satellite attitude movements during the viewing. On a short viewing, their errors are negligible, but on very long segments, a maximum error of 0.6 milliradians could happen, leading to a location error of 0.5 km between the first and last lines of the segment.

#### 2.2.2.5 Inter-camera misregistration

Because there are 4 distinct instruments in the VEGETATION payload, there are some differences between them due to optical and mechanical properties. These differences are less than a half-pixel (about 0.6 km), but they are corrected, because they were measured before launch. The resulting misregistration after correction is about 0.1 pixel.

#### 2.2.2.6 Geometrical modeling

The first step of the geometrical processing is to build 2 location grids referring to a “system projection”. This step, called geometrical modeling, produces two tables giving the location of every 8th pixel in column and row in a map reference that uses a specific map projection called “system projection”.

This map projection is in fact a set of 20 UTM projections tangential to the track of the satellite, chosen to fit best the given track. The location is given for 2 elevations (0 and 5000m); in such a way, the true elevation of the site will be taken into account at the last step of the processing, allowing to update the Elevation Model, without re-computing the locations grids. For that purpose, the satellite location is computed from an orbit bulletin (MADRAS) computed in the Spot control Centre, or from the on board navigator (DORIS) data, available in the telemetry, which gives directly the location of the satellite in an earth reference. Satellite attitude is estimated from the gyrometer data, which takes into account the satellite movements during the viewing.

The payload positioning on SPOT4 and the 4 cameras positioning on the payload produce a constant bias measured before launch that is taken into account.

For each pixel of the line, the viewing directions take into account lens characteristics, including lens distortion specific to each camera. These steps allow a virtual ray to be traced from the satellite to the earth surface, and the location of the viewed area corresponding to a given pixel to be determined.

This geometrical modeling is carried out for the near infrared band, the pixels of the other bands are located by difference to near infrared.

As a result, the Spot/VEGETATION images reach a geo-location accuracy of about 100m per 1km pixel. These images are chosen as reference images.

### 2.2.3 NDVI synthesis

NDVI synthesis is based on Maximum Value Composite, which is based upon the hypothesis that the best condition in a 10-day period of observation corresponds to the maximum value of NDVI, or that any source of noise leads to a drop in NDVI. The synthesis is done from global mosaics, after all geometric and radiometric correction, including atmospheric correction.

For a period of 10-day and for each pixel, the date corresponding to the maximum value of NDVI is kept in the final mosaic.

## 2.3 Projection of MSG images into Spot/VEGETATION grid

The MSG level 1.5 images are provided in geostationary satellite view projection. To combine these images with Spot/VEGETATION they must be reprojected into Spot/VEGETATION geodetic system (equi-rectangular geographic representation, WGS84 ellipsoid).

The MSG images are available in 4 windows (Figure 9). The file coordinates of each region with respect to the global view are given in (Table 3) [SAF/LAND/MF/PUM\_AL/1.4]:

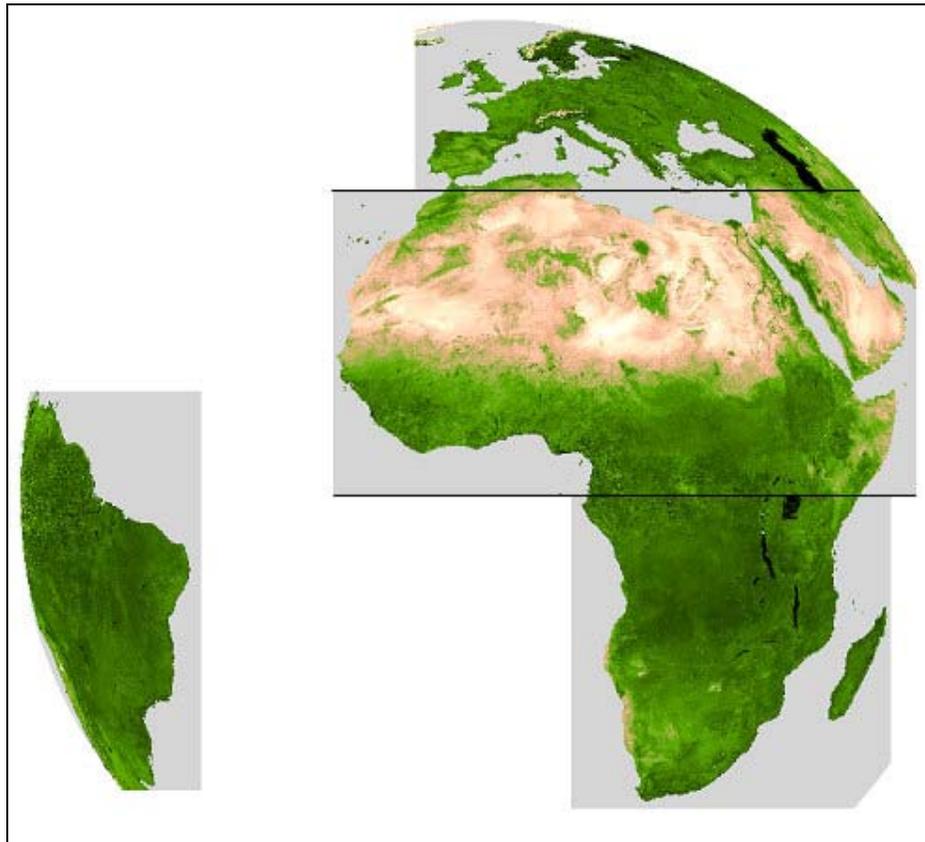


Figure 9. The LandSAF products are split on 4 windows: Europe (Euro), Northern Africa (NAfr), Southern Africa (SAfr), South America (SAm)

Region	Initial Column	Final Column	Initial Line	Final Line	Size in columns	Size in Lines
Euro	1550	3250	50	700	1701	651
NAfr	1240	3450	700	1850	2211	1151
SAfr	2140	3350	1850	3040	1211	1191
Same	40	740	1460	2970	701	1511

Table 3. Characteristics of the 4 LandSAF geographical areas. Each region is defined by the corner positions relative to an image of 3712 columns per 3712 lines with indices starting at 1 in the North and West [SAF/LAND/MF/PUM\_AL/1.4]

A python/gdal script was written to build the mosaic. The script syntax is the following:

Usage: mosaic\_msg.py [-of format] [-co options] [-nodata nodatavalue] [-nafr NorthAfricaFile] [-safr SouthAfricaFile] [-euro EuropeFile] [-same SouthAmerica] [-o output\_file]

The user can combine 1 to 4 input files into the final mosaic, each file corresponding to a region. The regions starting positions (Table 3) are stored in the script, the relation between an image and the position is done through the keywords (-nafr, -safr, -euro, -same). The script copies the data into an output file (name passed with option -o, file format by option -of, and file format option by -co option).

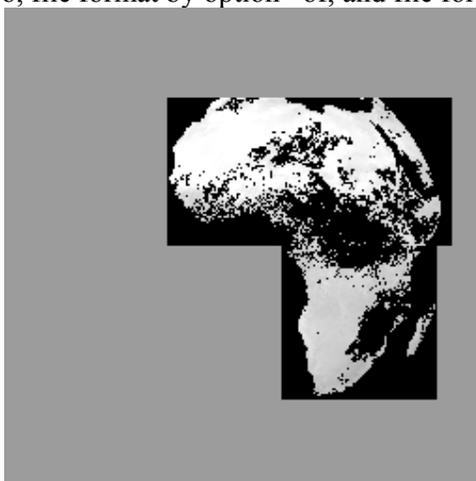


Figure 4. Mosaic of NAfr and SAfr regions done with mosaic\_msg.py

The second step consists in warping the image from global view to an equi-rectangular projection.

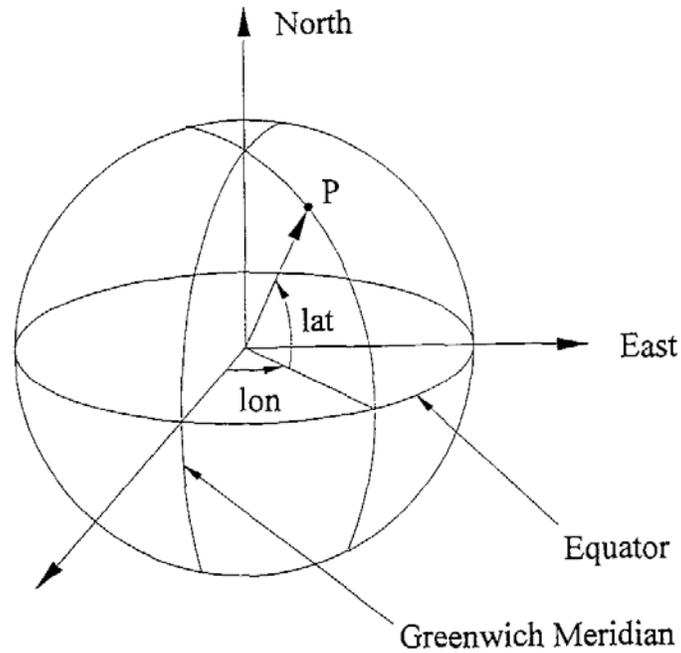


Figure 10. Geographical coordinates

The normalized geostationary projection describes the view from a virtual satellite to an idealized earth. Herein, the virtual satellite is in a geostationary orbit, perfectly located in the equator plane exactly at longitude  $sub\_lon$ . The distance between spacecraft and centre of earth is  $h=42164$  km. The idealized earth is a perfect ellipsoid with an equator radius of  $r_{eq}=6378.1690$  km and a polar radius of  $r_{pol} = 6356.5838$  km [CGMS/03]

The Earth is described as an oblate rotational ellipsoid.

Two cartesian coordinate frames are introduced. Coordinate system  $(e_1, e_2, e_3)$  has its origin in the centre of the earth. Vector  $(e_3)$  points in northern direction. Vector  $(e_1)$  points towards the Greenwich meridian. Coordinate system  $(s_1, s_2, s_3)$  has its origin at the satellite position. Vector  $(s_3)$  points northwards, and vector  $(s_1)$  directs to the centre of the earth.

We have:

$$\frac{e_3^2}{r_{pol}^2} + \frac{e_1^2 + e_2^2}{r_{eq}^2} = 1$$

The vector  $re$  points from the centre of the earth to a point P on the earth's surface. Thus,  $le$  is the longitude and  $fe$  is the geocentric latitude describing the point P. The transformation from geographic coordinates  $(lon, lat)$  is as follows:

$$\lambda_e = lon$$

$$\phi_e = \arctan \left[ \left( \frac{r_{pol}^2}{r_{eq}^2} \right) \times \tan(lat) \right]$$

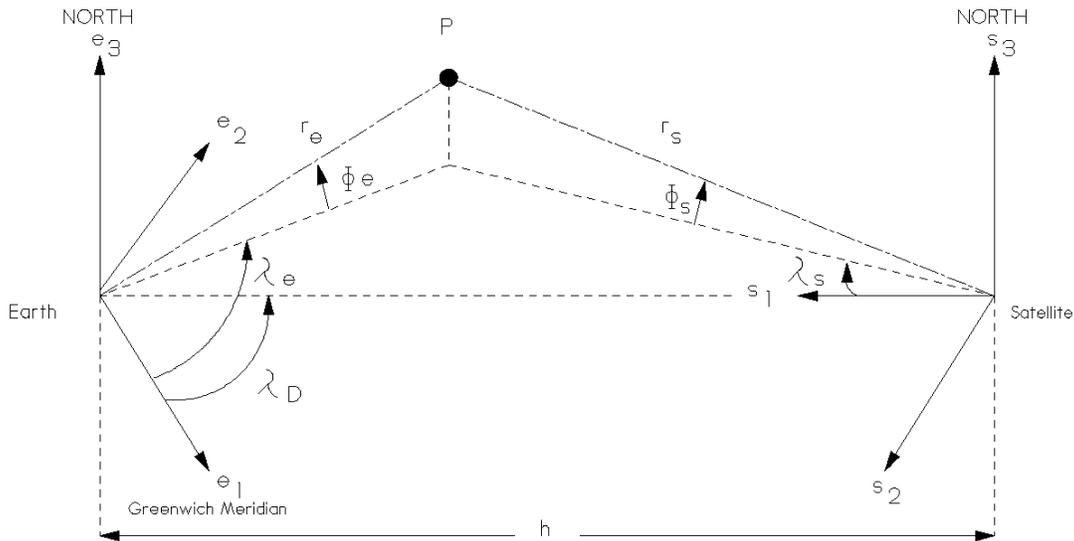


Figure 11. Coordinate frames for Geostationary Satellite Projection

The image coordinates are related to the latitude and longitude by the relation:

$$x = \arctan\left(\frac{-r_2}{r_1}\right)$$

$$y = \arcsin\left(\frac{-r_3}{r_n}\right)$$

With:

$$r_1 = h - r_1 \cos(c\_lat) \cos(lon - sub\_lon)$$

$$r_2 = -r_1 \cos(c\_lat) \sin(lon - sub\_lon)$$

$$r_3 = r_1 \sin(c\_lat)$$

$$r_n = \sqrt{r_1^2 + r_2^2 + r_3^2}$$

$$r_1 = \frac{r_{pol}}{\sqrt{1 - \frac{r_{eq}^2 - r_{pol}^2}{r_{eq}^2} \cos^2(c\_lat)}}$$

$$c\_lat = \arctan\left(\left(\frac{r_{pol}^2}{r_{eq}^2}\right) \tan(lat)\right)$$

The algorithm to project an MSG image, in geostationary view, into equi-rectangular projection, super-imposable to Spot/VEGETATION is the following:

- 1- define the output image size
- 2- for each pixel ( $x_{vgt}$ ,  $y_{vgt}$ ) of the output:
  - a. Compute latitude and longitude
  - b. Transform latitude and longitude into MSG pixel coordinate ( $x_{msg}$ ,  $y_{msg}$ )
  - c.  $Data[x_{vgt}, y_{vgt}] = data[x_{msg}, y_{msg}]$

The conversion of the SPOT/VEGETATION image coordinate, in an equi-rectangular projection, into latitude and longitude is straightforward:

$$lon = ULX + x * ps$$

$$lat = ULY - y * ps$$

The coordinates are defined at the centre of the pixel and the pixel size, ps, equals (1/112) °.

# 3

## Quality analysis of the re-projection

### 3.1 Consistency of projected and resampled MSG images with SPOT/VEGETATION

The simplest approach for projecting the MSG images consists in using the projections transformations described in section 2.3. This method guaranties having a projected image strictly matching the Spot/VEGETATION grids, which is mandatory for some types of processing involving Spot/Vegetation and Meteosat data.

Figure 12 shows the projected METEOSAT image of albedo in the red band (data source: landsaf) for date 2008/09/01 and the Spot/Vegetation 10 days synthesis in the red band for the time frame 2008/09/01-10. Only Northern Africa (Nafr) and Southern Africa (SAfr) were used (see definition on Figure 9).

The nature of the image shows some differences, especially on the cloudiest areas, such as the coast of the Gulf of Guinea, where the Spot/Vegetation shows significant differences with Meteosat data.

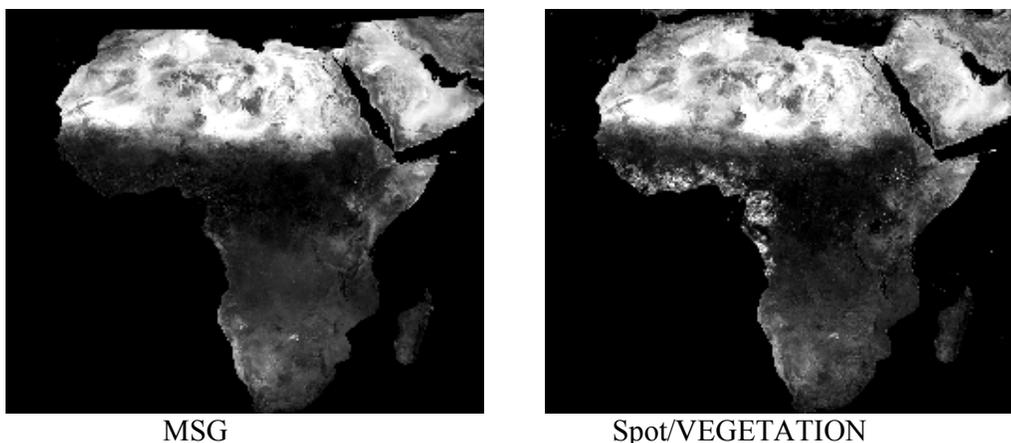


Figure 12. Projected MSG image and Spot/Vegetation (red bands).

METEOSAT and Spot/Vegetation are by nature two different systems. Spot/Vegetation is on a polar orbiting platform with progressive scanning which minimise image distortion, the pixel instantaneous field of view being almost constant across track. METEOSAT is a geostationary satellite which images distortion are those of the earth ellipsoidal shape, the pixel instantaneous field of view increasing with distance to the image centre (Figure 6 and Figure 7).

To evaluate the consistency of the METEOSAT projected images and the reference Spot/Vegetation images, the spatial correlation of images pairs were computed. The High-Accuracy Correlator software [Carmona-Moreno, 2003] performs this analysis by parsing the pair of images with sub-windows.

For a pair of sub-windows, the image correlation is computed. The sub-windows size was set to 40 pixels in order to contain geographic structures that can be compared on a pair of images (Figure 13). If the window was too small, the correlation between the two sources of images would be difficult to establish.

The image shifts are computed (x-offset, y-offset and norm) for each sub-window. Wherever a good correlation is found between the two images (i.e. the correlation software is able to retrieve similar objects in the same sub-window from both Spot and MSG), the images shifts provide a good estimate of the mis-registration error after the reprojection of MSG.

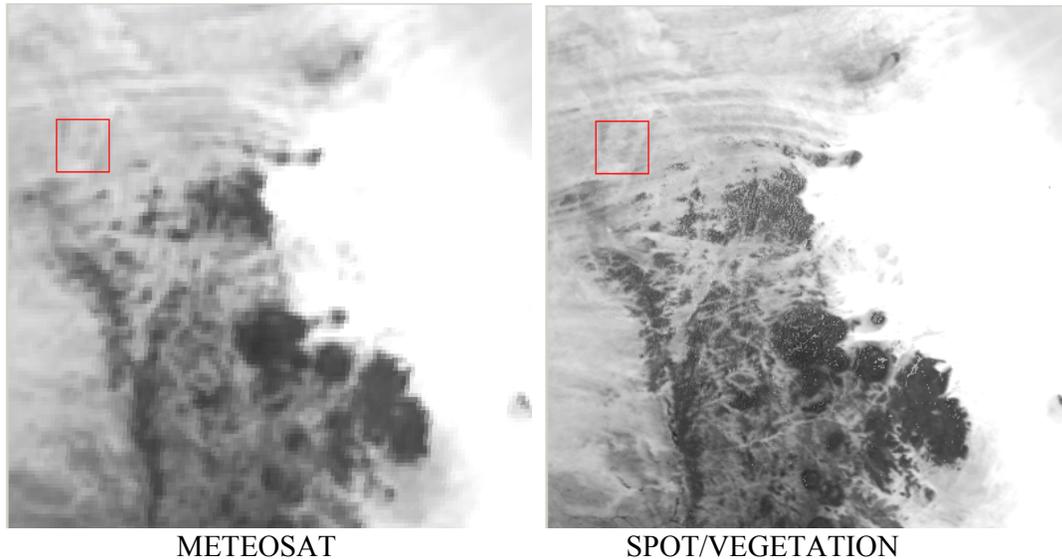


Figure 13. The correlation is analysed in windows of 40 pixels width (red box)

Correlation larger than 0.7 are arbitrarily set as good enough for assessing the images offsets. In arid and semi arid regions, good correlations are easily found (Figure 14). On dense forest and cloud screened regions, the correlation is more difficult or impossible to establish, like along the Gulf of Guinea or Nigeria and Chad. This lack of correlation is not related to the quality of the projection but much more on the properties of the landscape: continuous and homogeneous large fields are not appropriate for correlating the images.

Although the software estimates the image shift for each sub-window (Figure 15), these estimates are relevant only if the correlation between the two sub-windows is high enough. The largest differences are observed on the image edge (Southern Africa and Iran).

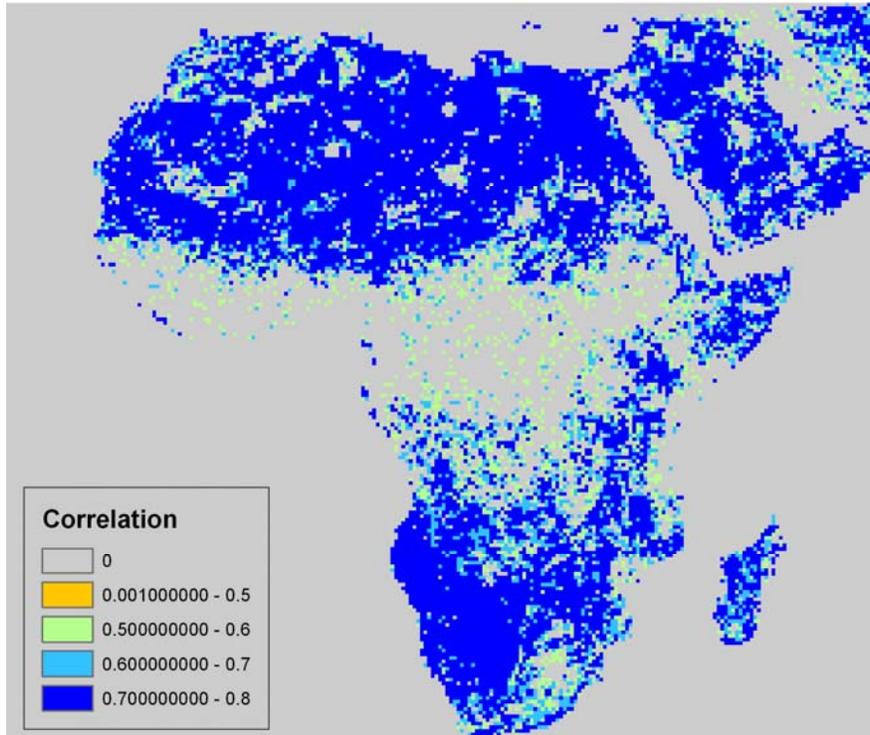


Figure 14. Areas with a spatial correlations better than 0.7 are good enough to estimate the images shifts.

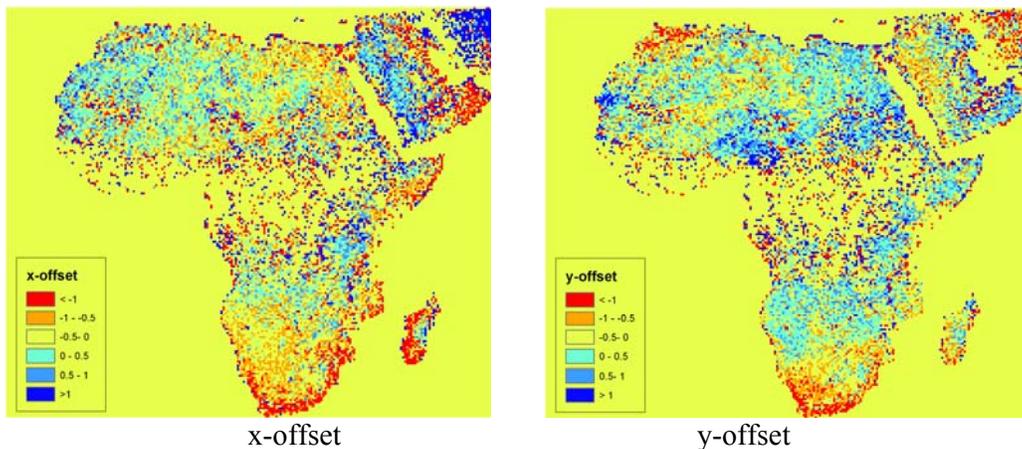


Figure 15. x and y offsets per sub-windows between Spot/Vegetation and Meteosat images.

The differences along the sea shore are explained by the sea masked used by Spot/Vegetation: a buffer of about 10 km is left, leading to inconsistencies with METEOSAT data and overestimated shifts.

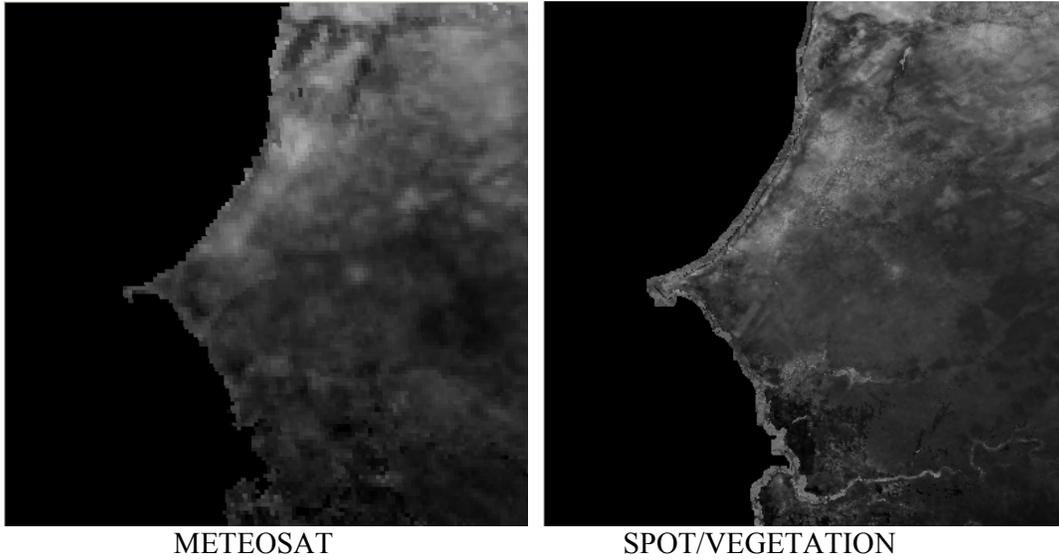


Figure 16. Spot/VEGETATION has a 10km buffer along the coast.

Nevertheless a large enough proportion of Africa is covered to estimate the images shifts (Figure 17, Figure 18 and Figure 19).

About 79% of the sub-windows show a shift lower than 1 pixel (1 km).

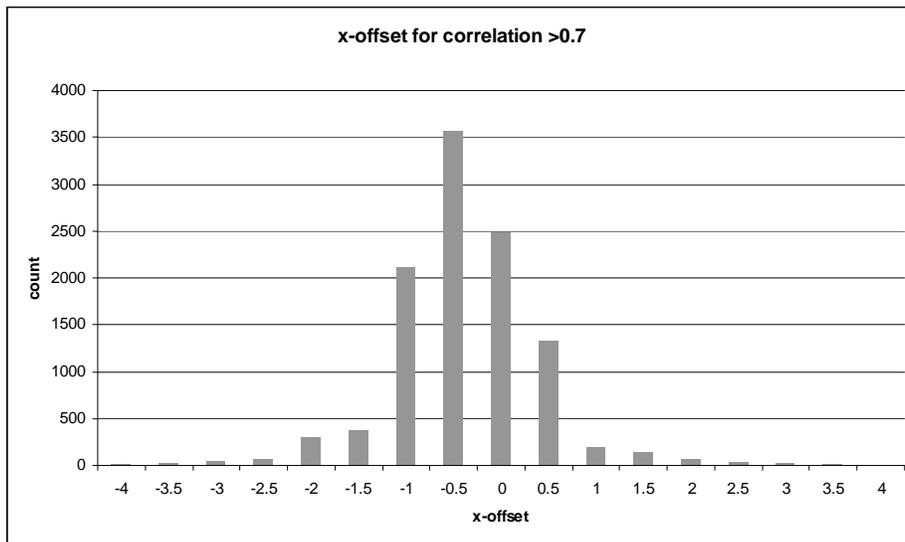


Figure 17. Sub-windows x- shifts, where correlation is better than 0.7

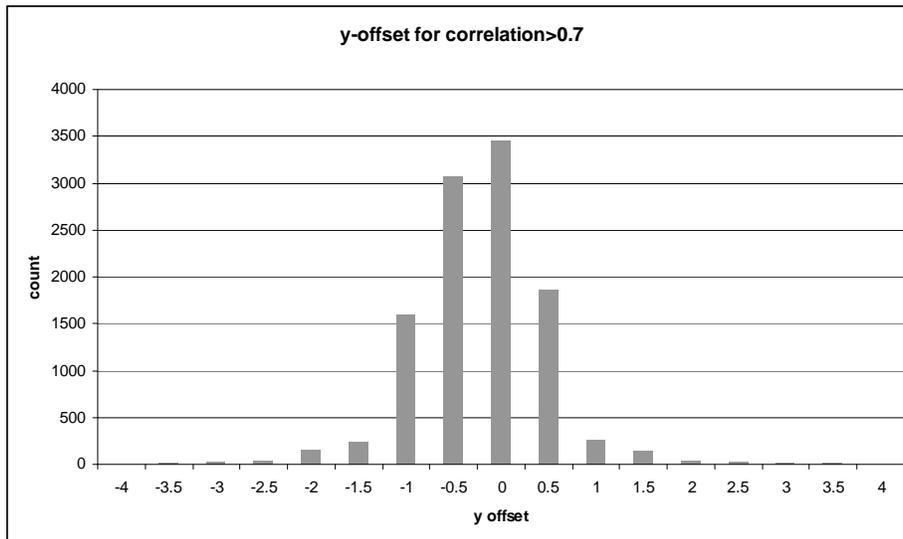


Figure 18. Sub-windows y-shifts, where correlation is better than 0.7

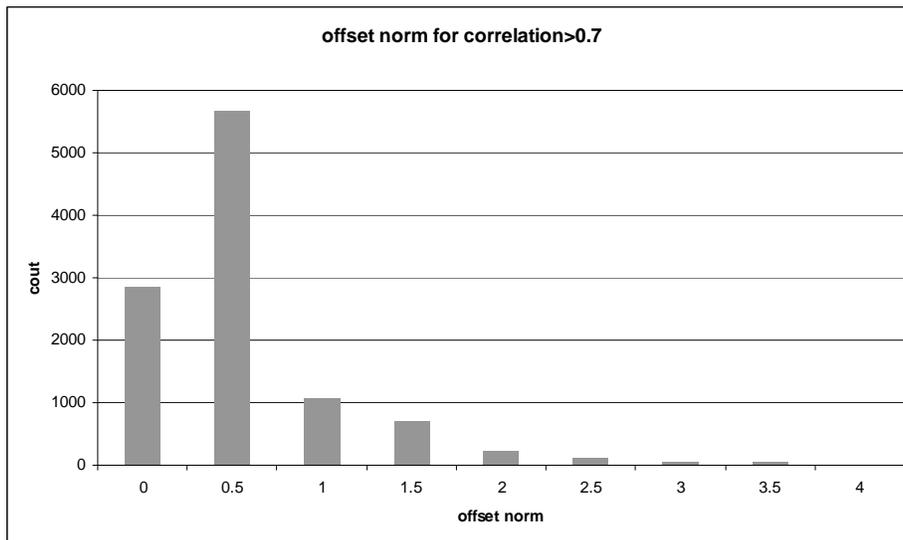
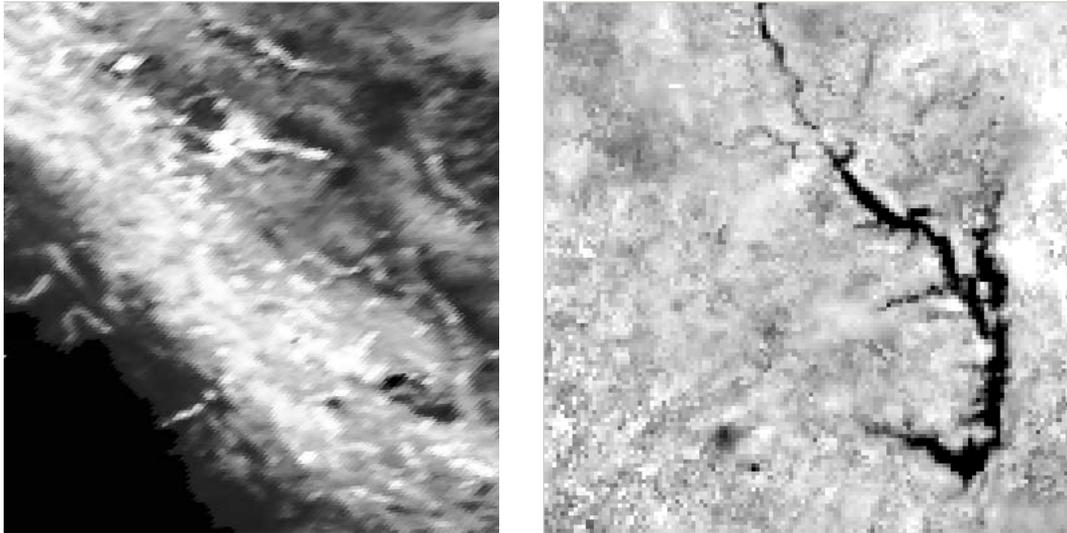


Figure 19. Sub-windows shift (shift norm), where correlation is better than 0.7

The re-projected images, sampled at 1km, give a good agreement with the reference SPOT/VEGETATION images. In addition, the pixel shifts are not too important on image edges. The re-projection of METEOSAT images and their re-sampling at 1km is found to be good, at least in term of geographical consistencies.

### 3.2 IFOV distortion

The projection and resampling of the MSG images to match the SPOT/VEGETATION grid is done on a nearest-neighbour basis. This choice allows faster projection processing, which can be critical when resampling at 1km. However, the drawback is that some image distortion is introduced, especially on the image edges, where the METEOSAT pixel covers larger instantaneous field of view.



Iran (image right corner)

Ghana (image centre)

Figure 20. Effect of pixel IFOV on MSG projected image

Figure 20 shows the instantaneous field of view distortion on the image border. The origin of this distortion is found in the geostationary satellite view: a pixel of the image located far from the centre (nadir) covers a larger ground surface. The re-projection algorithm does not include any re-sampling strategy, the resulting image having thus an aliased aspect on the borders. Around the centre of the image, the IFOV distortion is not too large, keeping the image actual resolution close to 3km.

### 3.3 METEOSAT band inter-registration and inter-correlation analysis

The image registration between bands was also evaluated. The approach is similar as the estimate of mis-registration between Spot/VEGETATION and MSG (see above).

The image correlation software was set with sub-windows of 20 pixels width: the images should be almost super-imposable with very similar features. The two visible bands C1 and C2 show the best correlations (Figure 21, Figure 22, null correlation sub-windows over the ocean are removed).

The measure of the distance (norm) between two channels show that a reasonable agreement is found between pairs of channels (Figure 23 and Figure 24).

The pixels misregistration is minimal between bands C1 and C2: for 83% of pixels in sub-windows showing a correlation better than 0.7, the measured translation is lower than 0.25 pixel, 12% of pixels show a misregistration between 0.25 and 0.5 pixel, the proportion dropping to 3% for a difference between 0.5 and 1 pixel.

The misregistration is slightly higher between bands C1 and C3: for 68% of pixels in sub windows showing a correlation better than 0.7, the measured misregistration is lower than 0.25pixel, and 24% show a misregistration between 0.25 and 0.5 pixel. The misregistration percentage drops to 5% for a difference between 0.5 and 1 pixel.

In conclusion, the MSG inter band registration is excellent, since it is found to be better than  $\frac{1}{4}$  pixel in most cases (for correlation above 0.7 in a given sub-window).

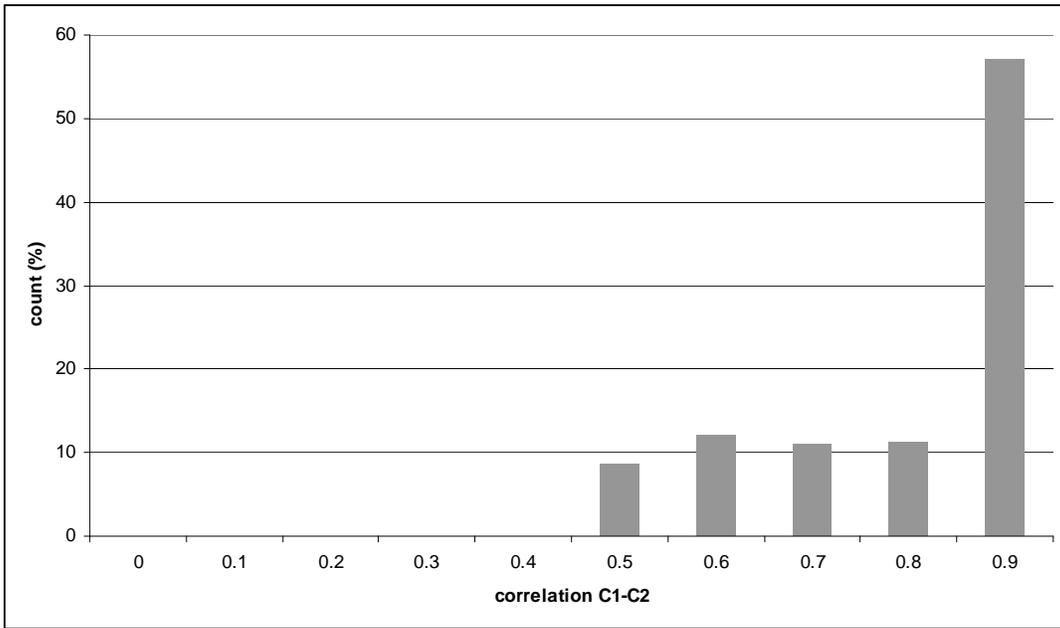


Figure 21. Correlations between bandsC1 and C2

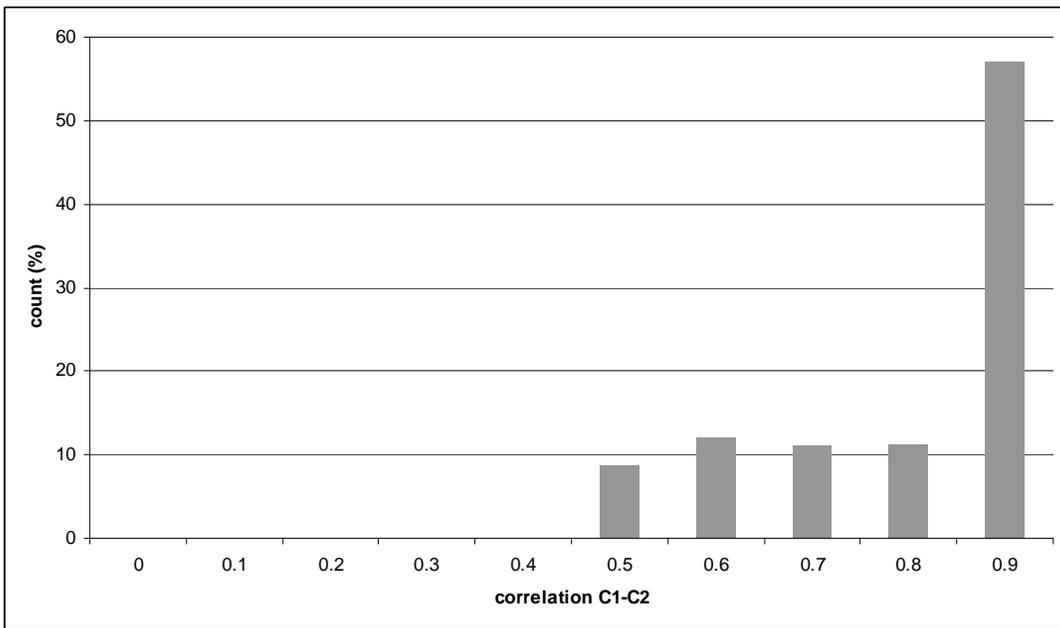


Figure 22. Correlation between bands C1 and C3

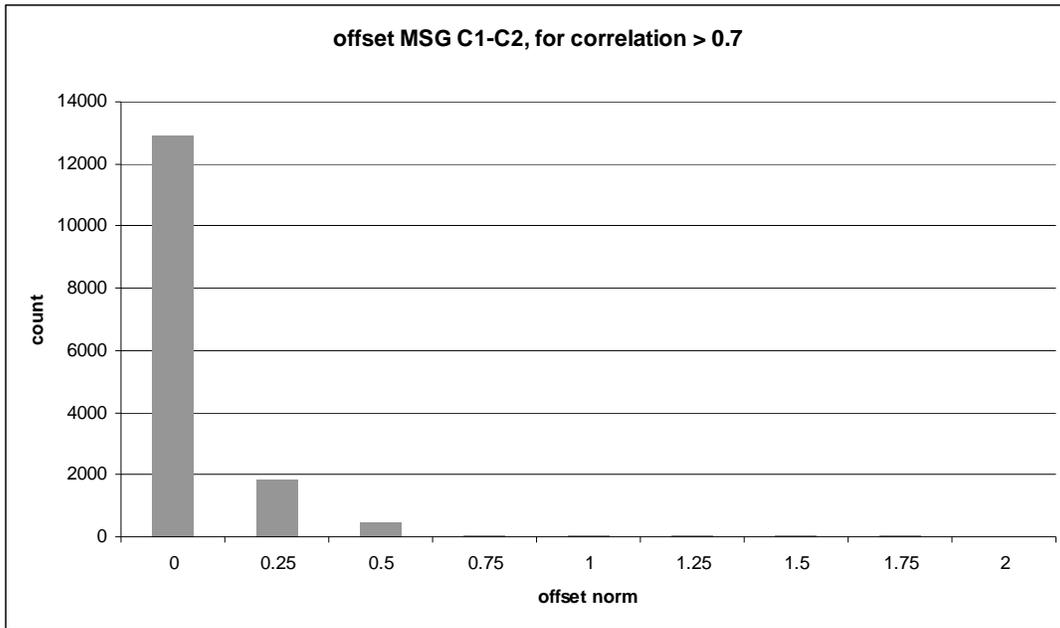


Figure 23. Windows shift (misregistration norm) between channels C1 and C2, where correlation is better than 0.7

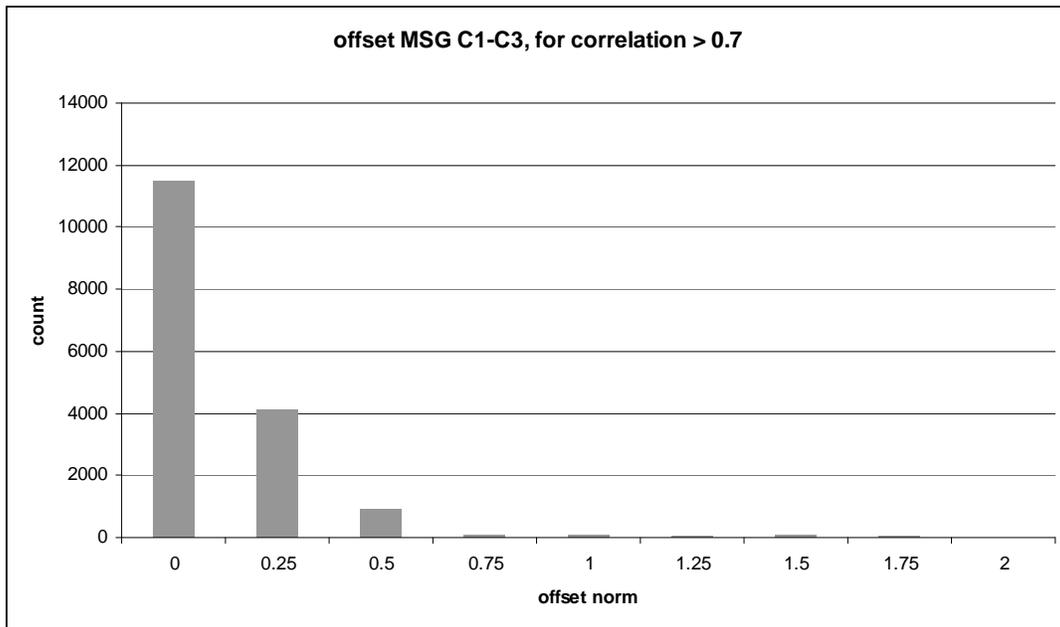


Figure 24. Windows shift (misregistration norm) between channels C1 and C3, where correlation is better than 0.7

# 4

## Conclusion

The EumetCast system offers to receive in near real time Earth observation from Spot/Vegetation and MSG/SEVIRI instruments, as well as many factors derived from these observations.

The algorithm and software described in this document show that the MSG observations and related factors can be projected on the Spot/VEGETATION projection system. The projected images show a sufficiently good geometric accuracy for merging information together with Spot/VEGETATION system over Africa. The best quality is found for Central Africa, while Western Africa, Southern and Eastern Africa showing more distortion.

# 5

## Annex

**Note: the software sources are given as an illustration of the text. The code in exploitation may differ from the source printed in this document.**

### 5.1 Mosaicing code

The purpose of the mosaicing code (`mosaic_msg.py`) is to assembly MSG regional sub window into a unique image having predefined dimensions (3712 x 3712).

This code is used to put together regional view and provide to the projection code an input which has predefined dimensions. This operation solve the problem

The code load `gdal`, `numpy`, `sys` and `math` libraries. To improve the computation time, `psyco` library is also loaded. Please note that this library working only on 32bits computer, it can be removed without prejudice to the code functioning.

```
#!/usr/bin/env python

# author: Bruno COMBAL, JRC, European Commission
# date: 4/12/2008
# purpose: reproject a GEOS image (MSG) into geographic

try:
    from osgeo import gdal
    from osgeo.gdalconst import *
    gdal.TermProgress = gdal.TermProgress_noch
except ImportError:
    import gdal
    from gdalconst import *
try:
    import numpy as N
    N.arrayrange = N.arange
except ImportError:
    import Numeric as N
try:
    from osgeo import gdal_array as gdalnumeric
except ImportError:
    import gdalnumeric

import sys
import math
try:
    import psyco
    psyco.full()
except ImportError:
    pass
```

The `mosaic` function gets on input an array of 4 strings, each of these strings can be either an empty one or a filename. The first string corresponds to Europe area image, the second one to Northern Africa, the third one to southern Africa, and the forth to South America. The corresponding images, when

define, must match the areas defined in section 2.3. The position of these images is predefined in array ROI (the positions start counting at 1).

The function also gets the output file name (outfile), its format as a gdal strings (GTiff for geotiff, HFA for Erdas Imagine, etc.) and format options as a string concatenating all the options.

Optionally, the user can define a no data value, which is used for defining the mosaic background (dataOut = N.ones((outNS, outNL)) \* nodata).

For each non null filename in fname, the code reads the file and copy it into dataout.

```
def mosaic_msg(fname, outfile, format, options, nodata):
    """ Copy input data into the geos view
    """
    # definitions: euro, nafr, safr, same
    # as defined in SAF/LAND/MF/PUM_AL/1.4, version 1.4, date 15/12/2006
    # on table 4, page 33
    # positions in the array start counting at 1
    roi=[[1550, 3250, 50, 700], \
         [1240, 3450, 700, 1850] , \
         [2140, 3350, 1850, 3040] , \
         [40, 740, 1460, 2970] ]

    roiView = [1, 3712, 1, 3712]
    outNS = roiView[1] - roiView[0] + 1
    outNL = roiView[3] - roiView[2] + 1

    # open files
    fid=[]
    ifile=[]
    dataTypeRef=None

    for ii in range(len(fname)):
        if (fname[ii] != ''):
            fidin = gdal.Open(fname[ii], GA_ReadOnly)
            fid.append(fidin)
            ifile.append(ii)
            dataType = fidin.GetRasterBand(1).DataType
            if (dataTypeRef is None):
                dataTypeRef = dataType
            elif (dataTypeRef != dataType):
                print "Files do not have the same type!"
                sys.exit(2)

    # output matrix dimensions
    dataOut = N.ones((outNS, outNL)) * nodata

    # total lines
    totallines=0
    for ii in fid:
        totallines=totallines+ii.GetRasterBand(1).YSize
    accumulines=0

    for ii in range(len(fid)):
        ipos = ifile[ii]
        inH = fid[ii].GetRasterBand(1)
        dataIn = inH.ReadAsArray(0, 0, inH.XSize, inH.YSize)
        initCol = roi[ipos][0] - 1
        lastCol = roi[ipos][1] - 1
        initLine = roi[ipos][2] - 1
        lastLine = roi[ipos][3] - 1

        for il in range(inH.YSize):
            for ix in range(inH.XSize):
                #print initCol + ix, initLine + il
```

```

        dataOut[initLine + il][initCol + ix] = dataIn[il][ix]

        gdal.TermProgress( (il + accumlines) / float(totallines))
        accumlines = accumlines + inH.YSize

# instantiate output file
outDriver = gdal.GetDriverByName(format)
outDS=outDriver.Create(outfile, outNS, outNL, 1, dataType, options)

# assume only 1 band
outband = outDS.GetRasterBand(1)

outband.WriteArray(gdalnumeric.array(dataOut), 0, 0)

gdal.TermProgress(1)

```

The code described above can be used as a library or be called from the command line. For this latter usage, a Usage function is implemented and called by the command line interface when an input error is detected. This subroutine stops the execution of the code (`sys.exit(1)`).

```

def Usage(message):
    print message
    print "Usage: mosaic_msg.py [-of format] [-co options] [-nodata nodatavalue]
[-nafr NorthAfricaFile] [-safr SouthAfricaFile] [-euro EuropeFile] [-same
SouthAmerica] -o output_file"
    print "Default nodata is set to 0"
    print
    sys.exit(1)

```

The main code itself parses the list of command line parameters and check its validity (Usage is called when an error is found, exiting the code).

The default output file format is geotiff. nodata is set to 0.

The code checks that the filenames passed in parameters actually exist on the file system. At least one file must be given to build the mosaic.

```
if __name__=="__main__":
    outfile = None
    format = 'GTiff'
    options = []
    nodata = 0
    regions=['',' ',' ',' '] # region order is: euro, nafr, safr, same

    ii = 1
    while ii < len(sys.argv):
        arg = sys.argv[ii]
        if arg == '-o':
            ii = ii + 1
            outfile=sys.argv[ii]
        elif arg == '-of':
            ii = ii + 1
            format=sys.argv[ii]
        elif arg == '-co':
            ii = ii + 1
            options.append(sys.argv[ii])
        elif arg == '-nodata':
            ii = ii + 1
            nodata = float(sys.argv[ii])
        elif arg == '-o':
            ii = ii + 1
            outfile = sys.argv[ii]
        elif arg == '-euro':
            ii = ii + 1
            regions[0] = sys.argv[ii]
        elif arg == '-nafr':
            ii = ii + 1
            regions[1] = sys.argv[ii]
        elif arg == '-safr':
            ii = ii + 1
            regions[2] = sys.argv[ii]
        elif arg == '-same':
            ii = ii + 1
            regions[3] = sys.argv[ii]
        ii = ii + 1

    if outfile is None:
        Usage('no output file defined, use -o option')
    if (regions[0]=='') and (regions[1]=='') and (regions[2]=='') and (regions[3]==''):
        Usage('Give at least one input file, -same -safr, -nafr, or -euro')

    print 'composition with ',
    if regions[0]!='':
        print 'euro=',regions[0],' ',
    if regions[1]!='':
        print 'nafr=',regions[1],' ',
    if regions[2]!='':
        print 'safr=',regions[2],' ',
    if regions[3]!='':
        print 'same=',regions[3],' '
    print

    mosaic_msg(regions, outfile, format, options, nodata)
```

To mosaic Nafr with SAfr images, into a geotif image with internal compress, the following command line can be used:

```
mosaic.py -of gtiff -co "compress=lzw" -nafr 200809010000_Nafr_AL-SP-BH.tif -safr 20080901000_SAfr_AL-SP-BH.tif -o mosaic_20080901000_AL-SP-BH.tif
```

## 5.2 Projection code

The purpose of the projection code (geos2geographic.py) is to transform a view of the earth image, stored in a 3712x3712 array into an image super imposable to Spot/Vegetation grid.

This codes loads the same libraries as the mosaic code.

```
#!/usr/bin/env python
# author: Bruno COMBAL, JRC, European Commission
# date: 4/12/2008
# purpose: reproject a GEOS image (MSG) into geographic

try:
    from osgeo import gdal
    from osgeo.gdalconst import *
    gdal.TermProgress = gdal.TermProgress_nocb
except ImportError:
    import gdal
    from gdalconst import *
try:
    import numpy as N
    N.arrayrange = N.arange
except ImportError:
    import Numeric as No
try:
    from osgeo import gdal_array as gdalnumeric
except ImportError:
    import gdalnumeric

import sys
import math
try:
    import psyco
    psyco.full()
except ImportError:
    pass
```

The main function, geos2geographics get the input file name (file), the output file name (outfile) as well as the output format and related options.

The target region of interest (t\_roiIN), the target pixel size (t\_psIN), the target pixel origin (usually 0.5 0.5 for pixel center definition).

Optionally, one can define the sat\_longitude (default is 0), and the nodata value. In addition, an image offset (s\_offIN) can be defined.

The function starts by defining the output: its size is given by the function make\_dims. The output has the properties of a spot/vegetation grids. That is the reason why the pixel size is corrected on the fly to be a multiple of 1/112.

Once the output array is defined, a loop goes through its dimensions, compute the corresponding lat/long (file2geo\_VGT) and transforms the coordinates into MSG file coordinates (geo2file) allowing to write the MSG pixel value into the output array.

```
def geos2geographic(file, outfile, format, options, s_offIN, t_roiIN, t_psIN, t_pixel_org,
sat_lon, nodataIN=0):

    """ Warp a Geostationary Satellite View projection into a geodetic system (WGS84).
    """
    # the input offset is a positive integer
    if s_offIN[0] < 0 :
```

```

        s_off[0]=int(0)
    else:
        s_off[0] = int(s_offIN[0])

    if s_offIN[1] < 0 :
        s_off[1]= int(0)
    else:
        s_off[1] = int(s_offIN[1])

    print "offset: ",s_off[0],s_off[1]

    # open input file
    fileFID = gdal.Open(file, GA_ReadOnly)
    dataType = fileFID.GetRasterBand(1).DataType
    # set the nodata to the right type
    nodata = (N.ones(1)*nodataIN).astype(type(dataType))
    # ensure the pixel size is a multiple of 1/112, at least 1/112
    # ensure pixel size is positive
    t_ps[0] = math.fabs((round(t_psIN[0] * 112))/112.0)
    t_ps[1] = math.fabs((round(t_psIN[1] * 112))/112.0)
    if (t_ps[0] != t_psIN[0]) or (t_ps[1] != t_psIN[1]):
        print "User pixel size ",t_psIN," corrected to a multiple of 1/112 as ",t_ps

    # arrange to have t_roi matching SPOT/VEGETATION grid
    t_roi[0] = math.floor( t_roiIN[0]/t_ps[0] ) * t_ps[0]
    t_roi[1] = math.floor( t_roiIN[1]/t_ps[1] ) * t_ps[1]

    # output matrix dimensions
    [outNS, outNL] = make_dims(t_roi, t_ps)

    # instantiate output file
    outDriver = gdal.GetDriverByName(format)
    outDS = outDriver.Create(outfile, outNS, outNL, 1, dataType, options)
    outDS.SetGeoTransform((t_roi[0]-math.fabs(t_pixel_org[0])*t_ps[0], t_ps[0], 0.0, \
        t_roi[1]+math.fabs(t_pixel_org[1])*t_ps[1], 0.0, -t_ps[1] ) )

    projection='GEOGCS["WGS84",DATUM["WGS_1984",SPHEROID["WGS
84",6378137,298.2572235629972,AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","6326"]],PRIMEM["G
reenwich",0],UNIT["degree",0.0174532925199433],AUTHORITY["EPSG","4326"]]'
    outDS.SetProjection( projection )

    # assume only 1 band
    outband = outDS.GetRasterBand(1)
    indata = fileFID.GetRasterBand(1)

    # read the whole input band
    dataIn = indata.ReadAsArray(0, 0, indata.XSize, indata.YSize)

    dataOut = gdalnumeric.array(N.zeros(outNS))
    dataOut.shape=[1, -1]

    print "Input image dimensions ", indata.XSize, indata.YSize
    print "Output image dimensions ", outNS, outNL
    print N.shape(dataOut)
    print N.shape(dataIn)

    for ilOut in range(outNL):
        gdal.TermProgress((ilOut+1)/float(outNL))
        for isOut in range(outNS):
            [lon, lat] = file2geo_VGT(isOut, ilOut, t_roi[0], t_roi[1], t_ps)
            [ixi, iyi] = geo2file_geos(lon, lat, sat_lon)

            ix = 3712 - ixi + s_off[0]
            iy = 3712 - iyi + s_off[1]
            # note that pixels out of coordinate system have ix=-1 iy=-1
            if (ixi==-1) or (iyi==-1) or (ix<0) or (iy<0) or (ix>=indata.XSize) or
(iiy>=indata.YSize):
                dataOut[0][isOut] = nodata
            else:
                dataOut[0][isOut] = dataIn[iy][ix]

```

```
outband.WriteArray(dataOut, 0, ilOut)
```

The MSG geographic coordinates (lat/long) into file coordinates, according to the definitions given in section 2.3.

```
def geo2file_geos(lon, lat, rflon=0):
#   ""
#   input
#   vlon: longitude
#   vlat: latitude
#   rflon: satellite longitude, default=0degrees
#   output xr: x position
#   yr: y position
#   ""
#geo2file_geos(lon, lat)
# Setup constants
re = 6378.160          # equatorial radius
h = 42164.0 - re      # Reference altitude
rp = 6356.5838
lpsi2 = 1             # spin direction

resol = 3712.0 # pixel/line number for MSG

deltax = 17.83 / resol # E-W scanning step  scan amplitude is 17.83 deg. for MSG
deltay = 17.83 / resol #

dtor = math.radians(1.0)
xlat = dtor * lat
xlon = dtor * lon
cosxlat = math.cos(xlat)
sinxlat = math.sin(xlat)
cosxlon = math.cos(xlon)
rom = ( re*rp ) / math.sqrt( rp*rp*cosxlat*cosxlat + re*re*sinxlat*sinxlat )
y = math.sqrt( h * h + rom * rom - 2.0 * h * rom * cosxlat * cosxlon )
r1 = y*y + rom*rom
r2 = h * h

if (r1 > r2):
    return [-1, -1]

rs = re + h
reph = re
rpqh = rp
coslo = math.cos(rflon * dtor)
sinlo = math.sin(rflon * dtor)
teta = math.atan( (rpqh/reph) * math.tan( xlat ) )
xt = reph * math.cos( teta ) * cosxlon
yt = reph * math.cos( teta ) * math.sin( xlon )
zt = rpqh * math.sin( teta )
px = math.atan( (coslo *(yt - rs*sinlo) - (xt - rs*coslo)*sinlo) / (sinlo*(yt -
rs*sinlo) + (xt - rs*coslo)*coslo) )
py = math.atan( zt * ( (math.tan(px)*sinlo - coslo) / (xt-rs*coslo)) *math.cos(px) )
px = px / dtor
py = py / dtor
xr = px / (deltax*lpsi2)
yr = py / (deltay*lpsi2)

if (xr >= 0.):
    xr = int( px / (deltax*lpsi2) ) + 0.5
else:
    xr = int( px / (deltax*lpsi2) ) - 0.5

if (yr >= 0.):
    yr = int( py / (deltay*lpsi2) ) + 0.5
else:
    yr = int( py / (deltay*lpsi2) ) - 0.5

xr = xr + 0.5*resol + 0.5
yr = yr + 0.5*resol + 0.5
```

```
return [int(xr), int(yr)]
```

The conversion of a spot/vegetation pixel coordinate into lat/long is trivial:

```
def file2geo_VGT(xx, yy, ulx, uly, ps):  
#   """Convert a pixel coordinate into longitude and latitude  
#   """  
    return (ulx + xx * ps[0] , uly - yy * ps[1])
```

Function `make_dims` computes the dimensions of the output region of interest:

```
def make_dims(roi, ps):  
    """Compute the dimension of an array corresponding to a geographic window  
    input: roi is a tuple (uly, uly, lrx, lry)  
    ps is the pixel size (psx, psy)  
    output: nx, ny  
    """  
    return(int(math.floor((roi[2]-roi[0])/ps[0]))+1, int(math.floor((roi[1]-roi[3])/ps[1]))  
+ 1 )
```

The code described above can be used as a library or be called from the command line. For this latter usage, a Usage function is implemented and called by the command line interface when an input error is detected. This subroutine stops the execution of the code (`sys.exit(1)`).

```
def Usage(message):  
    print message  
    print "Usage:\t\tgeos2geographic.py [-of format] [-co options]"  
    print "\t\t[-s_offset xoff yoff]"  
    print "\t\t[-t_ps psx psy] [-sat_lon sat_lon] [-nodata nodatavalue]"  
    print "\t\t-t_roi ulx uly lrx lry -t_pixel_org {0 0 / 0.5 0.5}"  
    print "\t\t-o output_file file"  
    print ""  
    print "Default output pixel size is 1./112. for each direction -t_ps .008928571428571  
.008928571428571"  
    print "-t_pixel_org is 0.5 0.5 or 0 0"  
    print "Default nodata is set to 0"  
    print ""  
    print "You can use one of -make_ref_file or -ref_file. "  
    print "-make_ref_file save the pixel positions in a file for the parameters you gave."  
    print "This means the created file is valuable for a given t_roi, t_ps sat_lon"  
    print  
    sys.exit(1)
```

The main code itself parses the list of command line parameters and check its validity (Usage is called when an error is found, exiting the code).

The default output file format is geotiff.

`t_roi` defines the output region

`-t_pixel_org` defines the pixel reference: 0 0 for the corner of the pixel, 0.5 0.5 for the center (which is the definition for spot/vegetation and msg). This parameter exists for exportation to other grids than spot/vegetation.

```
if __name__=="__main__":  
    outfile = None  
    format = 'GTiff'  
    options = []  
    file = None  
    t_ps = [4.0/112.0, 4.0/112.0]  
    t_pixel_org = []  
    t_roi= []  
    nodata = 0  
    sat_lon = 0  
    s_off=[0, 0]  
  
    ii = 1  
    while ii < len(sys.argv):  
        arg = sys.argv[ii]  
        if arg == '-o':  
            ii = ii + 1
```

```

        outfile=sys.argv[ii]
    elif arg == '-of':
        ii = ii + 1
        format=sys.argv[ii]
    elif arg == '-co':
        ii = ii + 1
        options.append(sys.argv[ii])
    elif arg == '-t_off':
        ii = ii + 1
        s_off[0] = float(sys.argv[ii])
        ii = ii + 1
        s_off[1] = float(sys.argv[ii])
    elif arg == '-t_ps':
        ii = ii + 1
        t_ps[0] = float(sys.argv[ii])
        ii = ii + 1
        t_ps[1] = float(sys.argv[ii])
    elif arg == '-t_roi':
        ii = ii + 1
        t_roi.append(float(sys.argv[ii]))
        ii = ii + 1
        t_roi.append(float(sys.argv[ii]))
        ii = ii + 1
        t_roi.append(float(sys.argv[ii]))
        ii = ii + 1
        t_roi.append(float(sys.argv[ii]))
    elif arg == '-nodata':
        ii = ii + 1
        nodata = float(sys.argv[ii])
    elif arg == '-t_pixel_org':
        ii = ii + 1
        t_pixel_org.append(float(sys.argv[ii]))
        ii = ii + 1
        t_pixel_org.append(float(sys.argv[ii]))
    elif arg == '-sat_lon':
        ii = ii + 1
        sat_lon = float(sys.argv[ii])
    else :
        file = arg
        ii = ii + 1

if file is None:
    Usage('input file is missing')
if outfile is None:
    Usage('no output file defined, use -o option')
if len(t_roi) != 4 :
    Usage('No output region defined, use -t_roi')
if len(t_pixel_org) != 2:
    Usage('Pixel origin (center or corner) is not defined. This parameter takes 2 val-
ues.')
```

geos2geographic(file, outfile, format, options, s\_off, t\_roi, t\_ps, t\_pixel\_org,
 sat\_lon, nodata)

To project a mosaic prepared with mosaic.py, the command lines may be like:

```

t_ps=$(echo "scale=14; 1.0/112.0" | bc)
geos2geographic.py -t_ps ${t_ps} ${t_ps} -t_roi -26 38 60 -35 -t_pixel_org 0.5 0.5 -o re-
proj_20080901_AL-SP-BH.tif mosaic_20080901000_AL-SP-BH.tif

```

which generates projected images similar to those used in this manual.

# 6

## References

[CGMS 03] Coordination Group for Meteorological Satellites, 1999, “LRIT/HRIT Global Specification”, pp. 53, Issue 2.6, 12 August 1999

[EUM/MSG/SPE/057] Eumetsat, 2006, “MSG Ground segment LRIT/HRIT mission specific implementation”, pp. 185, Issue 6, 21 June 2006.

[EUM/OPS/TEN/05/2562] Eumetsat, “Meteosat 8/MSG-1. Spectral response” , pp. 38

[EUM/OPS/TEN/05/2561] Eumetsat, “Meteosat-8/MSG 1 - Channel Spectral Bands”, pp. 2

[EUM/MSG/ICD/105] Eumetsat, 2007, “MSG Level 1.5 Image Data Format Description”, pp. 122, Issue v5A, 22 August 2007

[SAF/LAND/MF/PUM\_AL/1.4] LandSAF, “Product User Manual. Land Surface Albedo”, pp. 48

[SAF/LAND/IM/PUM\_LST/2.2] LandSAF, “Product User Manual. PUM LST. Land Surface Temperature“, pp. 49

[Lacaze and Berges, 2005] B. Lacaze, J-C. Berges, 2005 “Contribution of Meteosat Second Generation (MSG) to drought early warning”, Proc. Of the Int.l Conf.: Remote Sensing and Geoinformation in the assessment and monitoring of land degradation and desertification : state of the art and Operational perspectives. 7-9 Sept. 2005, Trier, Germany.

[Carmona-Moreno, 2003] Carmona-Moreno C., 2003, “High accuracy correlator”, Publications of the European Communities, EUR 20679 EN, Luxembourg: Office for Official Publications of the European Communities, [http://www-tem.jrc.it/PDF\\_publicis/2003/Carmona\\_Correlateur\\_EUR\\_03.pdf](http://www-tem.jrc.it/PDF_publicis/2003/Carmona_Correlateur_EUR_03.pdf)

[Wan and Dozier, 1996], Wan Z. and Dozier J., 1996, “A generalised split-window algorithm for retrieving landsurface temperature from space”, IEEE Trans. Geosci. Remote Sens., vol.34, no.4, pp. 892-905.

European Commission

**EUR 23945 EN – Joint Research Centre – Institute for Environment and Sustainability**

Title: Projection of Meteosat images into World Geodetic system WGS-84 matching Spot/Vegetation grid

Authors: Bruno COMBAL, Josué NOEL

Luxembourg: Office for Official Publications of the European Communities

2009 – 37 pp. – 21x 29.7 cm

EUR – Scientific and Technical Research series – ISSN 1018-5593

ISBN 978-92-79-12953-7

DOI 10.2788/2452

**Abstract**

Products derived from MSG/SEVIRI and Spot/VEGETATION are routinely provided to a large audience of users, i.a. through the EumetCast broadcasting system. Although MSG/SEVIRI has a significantly coarser resolution than Spot/VEGETATION, its thermal channels and higher rate of observation (every 15 minutes) extends the information extracted from Spot/VEGETATION images (1km resolution).

MSG/SEVIRI being a geostationary satellite and Spot/VEGETATION a polar orbiting system, the combination of both sources of information requires to re-project the images in a unique cartographic system. This document shows how to project MSG images onto the Spot/VEGETATION grid, in a WGS84 plate carrée representation. The document also analyses the image distortion of the geostationary image and of its reprojection.

The results have a sufficient geometric quality to allow the combination of re-projected MSG images with Spot/VEGETATION S10 products.

### **How to obtain EU publications**

Our priced publications are available from EU Bookshop (<http://bookshop.europa.eu>), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents. You can obtain their contact details by sending a fax to (352) 29 29-42758.

The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.

LB-NA-23945-EN-C

