RMIB GERB Processing: Resolution Enhancement

Steven Dewitte, Luis Gonzalez, Aline Hermans

23rd May 2006

This document is outdated and does not represent the current state of the GERB processing.

Contents

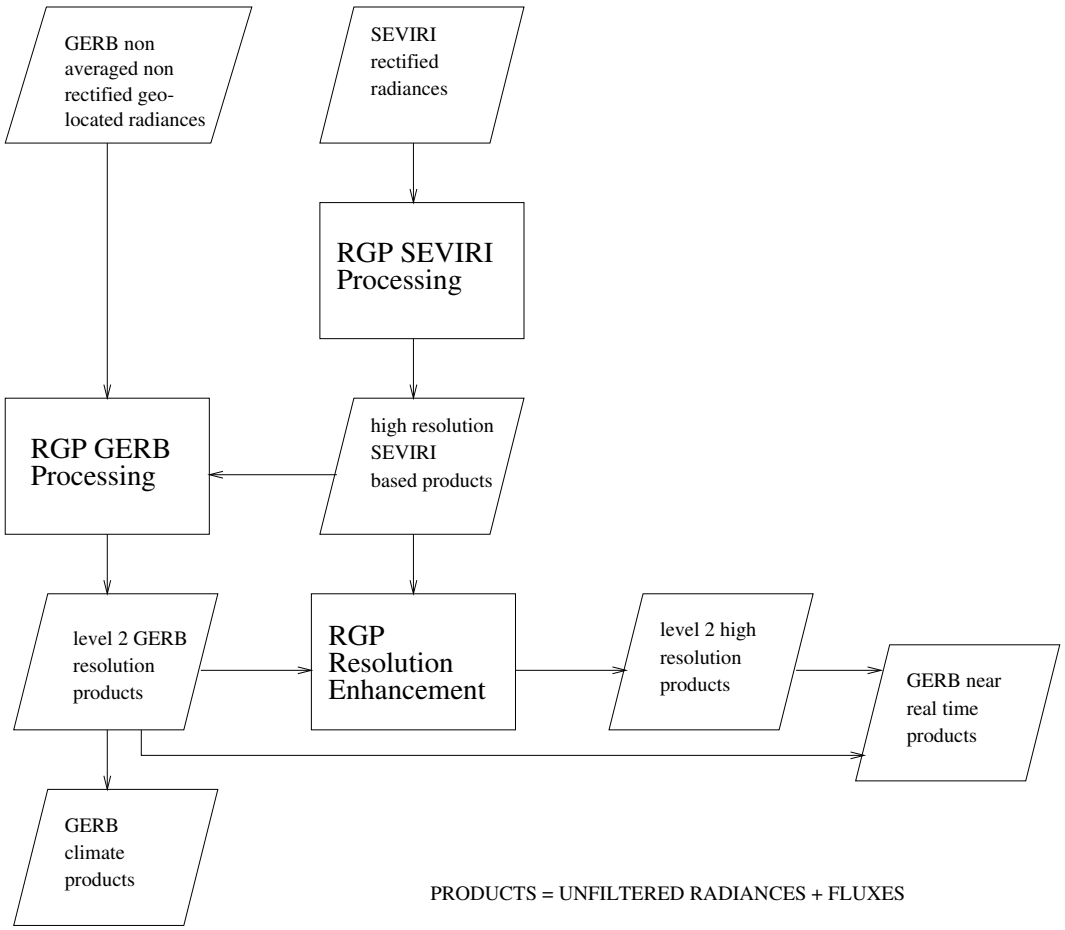


Figure 1: Division into susbsystems of the RMIB GERB Processing. The three blocks 'RGS SEVIRI Processing', 'RGS GERB Processing' and 'RGS Resolution Enhancement' are executed in sequential order.

1 Introduction

1.1 Scope of this document

This document gives a development strategy overview of the subsystem "Resolution Enhancement" of the RMIB part of the GERB Ground segment Processing (RGP), and describes briefly the final implementation.

Final implementation based on several tests on simulated and real Eumetsat images, is described in annexes.

1.2 Role of the Resolution Enhancement within the RGS

The RMIB part of the GERB ground segment consists in total of three subsytems: 'RMIB SEVIRI procesing', 'RMIB GERB processing' and 'Resolution Enhancement', to be executed in sequential order. An RGP overview is described in more detail indocument MSG-RMIB-GE-TN-0004[4]. See also figure 1.

The subsytem 'RGS SEVIRI Processing' has as input full resolution SEVIRI spectrally narrowband radiances and as output SEVIRI pixel resolution filtered radiance estimates and broadband unfiltered estimates. Those outputs are referred to as 'high resolution SEVIRI based

products'[?][6][7].

The subsystem 'RGS GERB Processing' has as input the GERB measured filtered broadband radiances as well as the high resolution SEVIRI based products, and as output GERB broadband unfiltered fluxes. Those outputs are referred to as 'level 2 GERB resolution fluxes'[5].

The subsystem ' Resolution Enhancement' has as input the high resolution SEVIRI based flux estimates and the level 2 GERB resolution fluxes and as output high resolution fluxes that are compatible with the level 2 GERB resolution fluxes. Those outputs are referred to as 'level 2 high resolution fluxes'.

1.3 Description of the Resolution Enhancement problem (or processing requirements)

The purpose of the RE process is to enhance the level 2 GERB fluxes resolution (GxG=256x256 pixels) to the high resolution SEVIRI (SxS=1233x1233 pixels) by use of SEVIRI fluxes estimation.

Let's denote :

1. $\widetilde{F}_{HR}(i,j)$ the high resolution SEVIRI based flux estimates, issued from the subsytem 'RGS SEVIRI Processing', and defined at each SEVIRI pixel (i,j),
2. $F_{LR}^{L2}(x,y)$the low resolution GERB L20-product flux, issued from the subsystem'RGS GERB Processing', and defined at each GERB pixel (x,y),
3. $P^d(i,j)$ the Point Spread Function (PSF)at pixel (i,j) for detector d.

In an ideal case, the down sampling of the high resolution measurement, weightened by the PSF, should reproduce the low resolution measurement. We should have :

$$F_{LR}^{L2}(x,y) = \sum_{i_{xy}} \sum_{j_{xy}} [P^d(i,j)\, \widetilde{F}_{HR}(i,j)]$$

where $i_{xy}, j_{x,y}$sweep the PSF area ({x-P,x+P} and {y-P,y+P}).

In reality this equality is not verified, because the calibration quality, and the scanning time interval, of both instruments, GERB and SEVIRI, are different.

Forgetting time correlation problem that will be discussed in the next section1.4, the main aim of the RE processing will be to find correction factor $cS(i,j)$ to the high resolution SEVIRI based flux estimates $\widetilde{F}_{HR}(i,j)$, so that the integration of the corrected flux estimates does reproduce the low resolution GERB L20-product flux $F_{LR}^{L2}(x,y)$. We would like to satisfy, at each GERB pixel (x,y),the condition :

$$F_{LR}^{L2}(x,y) = \sum_{i_{xy}} \sum_{j_{xy}} [P^d(i,j)\, cS(i,j)\, \widetilde{F}_{HR}(i,j)]$$

These conditions can be seen as G^2linear equations (one for every low resolution pixel(x,y)) in the S^2unknowns $cS(i,j)$ (one unknown for every high resolution pixel (i,j)). Clearly, this set of equations has more than one possible solution. In order to obtain a practical solution, different strategy were designed (see section 2).

1.4 Time concern in the Resolution Enhancement.

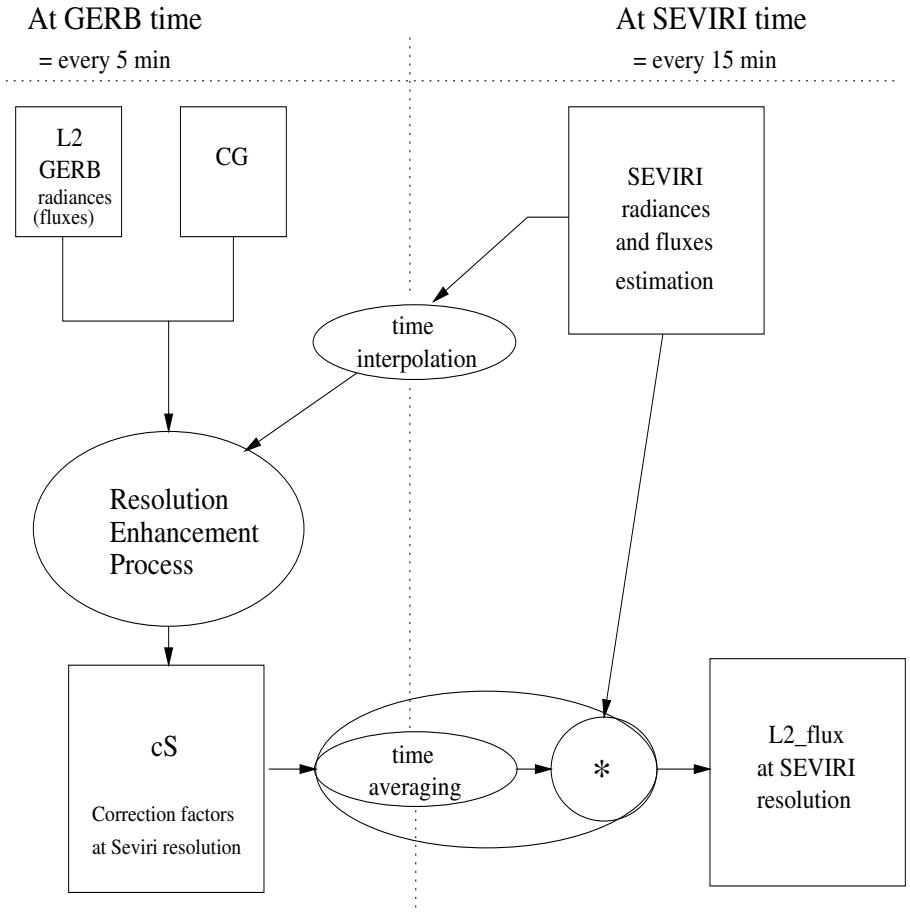
GERB and SEVIRI instruments do not scan Earth neither at the same time intervals, nor in the same way. The GERB instrument takes 2 images of the Earth (1 Short Wave and 1 Total Wave) each 5 minutes 14 secondes, columns by columns, while SEVIRI instrument scans Earth, 3 lines by 3 lines, each 15 minutes. Consequently, scanning time according to SEVIRI based data $\widetilde{F}_{HR}(i,j)$ and GERB based data $F_{LR}^{L2}(x,y)$ are different. It's necessary to temporally match these data before any comparison or combination. Since we decide to execute the core of the Resolution

Enhancement at GERB acquisition time, this core process will be preceeded and followed by time matching operations.(see next scheme)

First, SEVIRI data are interpollated at GERB acquisition time $t = t_G$.

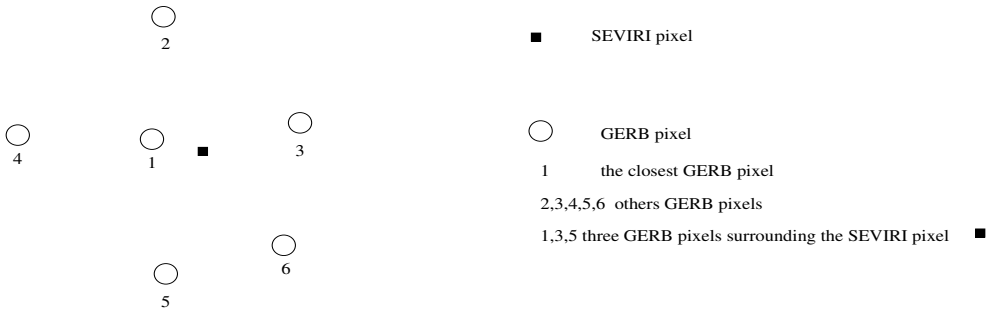
Secondly, the main job of the resolution enhancement, the estimation of correction factor is computed at GERB acquisition time.

Third, in order to produce a final output according to SEVIRI acquisition time, a weighted average for a 15 minutes interval centered on the SEVIRI acquisition time t_S, is worked out. Actually, only the correction factor is averaged, and then multiplied to the high resolution SEVIRI based flux estimation at SEVIRI acquisition time t_S. So the resulting corrected high resolution flux is a 'snapshot' flux instead of a real average flux.



- First, SEVIRI data are interpollated at GERB acquisition time $t = t_G$.
- Secondly, the main job of the resolution enhancement, the estimation of correction factor is computed at GERB acquisition time.
- Third, in order to produce a final output according to SEVIRI acquisition time, a weighted average of the correction factor for a 15 minutes interval centered on the SEVIRI acquisition time t_S, is worked out.
- Finally, corrected (or level 2) high resolution flux according to SEVIRI acquisition time is obtained from multiplication of the time averaged correction factor by the input high resolution SEVIRI based flux estimation.

Figure 2: Triangular geolocation-based interpolation.



Actually, time interpolation of SEVIRI based flux estimation at SEVIRI time to GERB time is already computed and saved on disk in the subsystem "RMIB GERB processing". This step is, then, out of concern in the RE subsystem.

We think more practical to divide the RE subsystem in 2 processes :

1. The first process is the core of the Resolution Enhancement system, described in section4, and will give as output the high (or SEVIRI) resolution correction factor at GERB time : $cS(i,j,t^g)$.
2. The second process, described in section 5, will average these estimations at GERB time to SEVIRI time, and then compute the final output, the L2 high resolution corrected flux averaged at SEVIRI time $F_{HR}^{L2}(i,j,t^s)$.

2 PROCESSING DEVELOPMENT STRATEGY

2.1 Linear system resolution approach

The first resolution approach is to convert conditions1.3 into a set of G^2equations in G^2 unknowns. Following this idea, we introduce, $CG(x,y)$, one unknown correction factor per GERB pixel (x,y). And we impose as extra condition on the high resolution correction factors $cS(i,j)$ they are interpolated values of the low resolution correction factors $CG(x,y)$:

$$cS(i,j) = \sum_{k=1}^{3} I_\Delta(i,j,x_k,y_k)\, CG(x_k,y_k) = \sum_{x'}\sum_{y'} I_\Delta(i,j,x',y')\, CG(x',y')$$

As a baseline, we use a triangular geolocation-based interpolation function : $I_\Delta(i,j,x',y')$.

More precisely, correction factor at SEVIRI pixel (i,j) is an interpolation of correction factor values at 3 GERB pixels $((x',y') = (x_k,y_k)\, with\, k = 1,2,3)$ surrounding, in geolocation terms, the considered SEVIRI pixel. The 3 GERB pixels $((x_k,y_k)\, with\, k = 1,2,3)$ are chosen among 5 GERB pixels : the closest (in geolocation terms) pixel, and its 4 direct neighbours.(see fig 2)

Substituting the extra condition in the undetermined set of equations, we obtain a linear system of G^2equations in the G^2 unknowns $CG(x,y)$.

$$F_{LR}^{L2}(x,y) = \sum_{i_{xy}}\sum_{j_{xy}} \left[P^d(i,j) \left(\sum_{x'}\sum_{y'} I_\Delta(i,j,x',y').CG(x',y') \right) \widetilde{F}_{HR}(i,j) \right]$$

This set of equations can be written in matrix form as $F = MC$, where F is the vector of elements $F_{LR}^{L2}(x,y)$, C is the vector of elements $CG(x,y)$, and M is the matrix of elements :

$$M[(x,y),(x',y')] = \sum_{i_{xy}} \sum_{j_{xy}} \left[P^d(i,j)\, I_\Delta(i,j,x',y')\, \widetilde{F}_{HR}(i,j) \right]$$

From a theoretical point of view, the vector C can be obtained by inverting M matrix : $C = M^{-1}F$

Once the low resolution factors $CG(x,y)$ are known, the high resolution factor $cS(i,j)$ can be calculated using interpolation relation 2.1.

But from a practical point of view, full storage and explicit inversion of matrix M (of size $256^2 * 256^2 > 4.10^9$) is unconceivable.

Looking forward to the M coefficients, we remark M is a matrix with a diagonal band structure. The resolution of $C = M^{-1}F$ seen as a diagonal system of linear equations, using LU decomposition[2], was tested at lower scale (i.e. with $G^2 = 4000$ unknowns). The computing time, of about 30 minutes at this lower scale, demonstrates us extrapolation to real scale system is barely conceivable.

As an alternative approach, we will try to compute the corrections $CG(x,y)$and $cS(i,j)$ by an iterative process.

2.2 Iterative approach

Aiming to solve the linear system $F = MC$ by avoiding explicit inversion of matrix M, a iterative approach is followed. At the k'th step of the iteration one has the approximations $CG(x,y) \cong CG^{(k)}(x,y)$ and $cS(i,j) \cong cS^{(k)}(i,j)$

Condition 1.3 can be re-written in terms of the sum of the current approximation plus a needed extra correction :

$$F_{LR}^{L2}(x,y) = \sum_i \sum_j [P^d(i,j)\, cS^{(k)}(i,j)\, \widetilde{F}_{HR}(i,j)] + \sum_i \sum_j [P^d(i,j) \left(cS(i,j) - cS^{(k)}(i,j) \right) \widetilde{F}_{HR}(i,j)]$$

Let's assume :
$cS(i_{xy}, j_{xy}) = \sum_{k=1}^{3} I_\Delta(i,j,x_k,y_k)\, CG(x_k,y_k) \cong CG(x,y)$
where $(x,y)$is the GERB pixel where the condition is considered.
The needed extra correction can be developed ;
$\sum_{i_{xy}} \sum_{j_{xy}} [P^d \left(cS - cS^{(k)} \right) \widetilde{F}_{HR}]$
$= \left(CG(x,y) - CG^{(k)}(x,y) \right) \sum_{i_{xy}} \sum_{j_{xy}} [P^d(i,j)\, \widetilde{F}_{HR}(i,j)]$
After substitution, we find the iteration equation :

$$CG^{(k+1)}(x,y) = CG^{(k)}(x,y) + \frac{F_{LR}^{L2}(x,y) - \sum_i \sum_j [P^d(i,j)\, cS^{(k)}(i,j)\, \widetilde{F}_{HR}(i,j)]}{\sum_i \sum_j \left[P^d(i,j)\, \widetilde{F}_{HR}(i,j) \right]}$$

The iteration recipe is as follows :

1. Start with $CG^{(0)}(x,y)$ and $cS^{(0)}(i,j)$= interpolation of the $CG^{(0)}(x,y)$.
2. Apply the iteration equation 2.2 to calculate $CG^{(k+1)}(x,y)$ from $CG^{(k)}(x,y)$ and $cS^{(k)}(i,j)$.
3. Calculate $cS^{(k+1)}(i,j)$ by interpolating $CG^{(k+1)}(x,y)$.
4. Repeat the second and the third step until convergence criterion is reached.

Two iterative schemes were developed and tested, but have given the same disappointing results: In the first algorithm, there is not storage of M matrix. The corrected flux convolution is explicitely computed, at each iteration, for every GERB point. In the second algorithm, the M coefficient are calculated and stored before the first iteration, then iterative correction of vector C, using pre-conditioning matrix method, try to reach a better C estimation.

For both algorithm, the iterations stop criterion is to reduce the maximal error on correlation condition 1.3 below 1% :

$$MAX_{x,y}\left|F_{LR}^{L2}(x,y)-\sum_{i_{xy}}\sum_{j_{xy}}[P^d(i,j)\,cS(i,j)\,\widetilde{F}_{HR}(i,j)]\right|<0.01$$

Tested on simulated images (of size 900x900 for SEVIRI and 64x64 for GERB), these algorithms lead to good results if the PSF area is smaller than 14 SEVIRI pixels, i.e. if there is not overlapping of the influence of the low resolution factor $CG(x,y)$ to the high resolution factor $cS(i,j)$. But, for PSF area of side size bigger than 14 (tested with 41), both iterative scheme versions start with reducing error during the firsts iterations, then diverge dangerously, far from the 1% error requirement. Unfortunately, for the real GERB image, the PSF area overlaps more than one GERB geolocation.

2.3 Lagragian approach

As the two first approaches were unsucceeful, we will try to design the Resolution Enhancement problem as Generalised Lagrange function to minimise under 'm' constraints. This approach is, first, described and tested on simplified problem (see section 3). Successfull results encourage us to implement the Lagragian methods on full or "real" problem (see section 4)

3 METHODOLOGY DESCRIPTION AND TESTING THROUGH SIMPLIFIED APPROACH

3.1 Simplified Problem Description

For the methodology explanation, only a simplified version of the problem is considered. It is expected that the essential features of the problem have been taken into account in order to allow extrapolation to the real problem.

The original data are two images: a high resolution image (HRI) and a low resolution image (LRI). The LRI is a low resolution version of the HRI. The two images are related by the simple window average:

$$L(x,y)=\frac{1}{(2N_c+1)(2N_l+1)}\sum_{k=-N_l,..,N_l,l=-N_c,..,N_c}c(xS_l+k,yS_c+l)H(xS_l+k,yS_c+l)\quad(1)$$

where N_c (N_l) is the window half-height (half-width), S_l (S_c) is the sampling distance between lines (columns) and c is a correction factor that is unknown.

The correction factors c are unknown and have to be determined. They are not unique since the problem is underdetermined (only one equality equation of type 1 for $S_c{*}S_l$ correction factors[1]). So additional features must be added to define what can be considered as a good solution. Constraints in order to have a smooth correction image will be shown to be a good candidate. As simple smoothing constraints:

$$\frac{1}{2}\sum_{i,j}(c(i,j)-\frac{1}{8}\sum_{k,l=-1,..,1\neq\{0,0\}}c(i+p,j+q))^2\quad(2)$$

This constraints means that a correction factor must be the mean of its 8 neighbours. The solution of this kind of constraints is a linear variation of the correction factors.

[1] The solution is trivial if the windows are not over-lapping. It is always considered that the windows overlap.

3.2 Lagrange Minimization Methodology

Let $f(x)$ a function to minimise under m constraints $g_i(x) = 0$. This problem can be numerically solved with a simple minimization method on the generalized Lagrange function (see [1]):

$$\Lambda(x, \lambda, r) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + r \sum_{i=1}^{m} [g_i(x)]^2 \tag{3}$$

The λ parameters are the usual Lagrange parameters. The value of r can be chosen freely. This last term takes into account the constraints even when the Lagrange parameters are null (see below the first step of numerical solution).

The numerical iterative solution is the following:

1. start with λ null (or positive)
2. minimise the function 3 for x
3. update the Lagrange parameters with an iteration towards the maximum for λ, i.e. $\lambda^{k+1} = \lambda^k + \rho_k g(x^k)$ where $g(x^k)$ is the gradient for the Lagrange parameters and ρ_k depends on the minimization method.
4. if stop test is verified end else go to step 2

3.3 Application on simplified problem

3.3.1 Generalized Lagrange Function

For the considered problem, the Generalized Lagrange function is:

$$\Lambda(c, \lambda, r) = \frac{1}{2} \sum_{i,j} [\overline{C}(i,j)]^2 + \sum_{x,y} \lambda_{x,y} E(x,y) + r \sum_{x,y} [E(x,y)]^2$$

where

- (i, j) are the pixel coordinates in HR image,
- (x, y) are the pixel coordinates in LR image,
- $\overline{C}$ is the local difference between a value and its neighbours:

$$\overline{C}(i,j) = c(i,j) - \frac{1}{8} \sum_{k,l=-1,..,1 \neq \{0,0\}} c(i+k, j+l)$$

- and the constraints are

$$E(x,y) = \frac{1}{(2N_c+1)(2N_l+1)} \sum_{k,l=-N,..,N} c(xS_l+k, yS_c+l) H(xS_l+k, yS_c+l) - L(x,y)$$

3.3.2 Iterative equation

The gradient of correction factor $c(p,q)$ is for the first term of $\Lambda(c,\lambda,r)$:

$$\delta_1 = \overline{C}(p,q) - \frac{1}{8} \sum_{m,n=-1,..,1\neq\{0,0\}} \overline{C}(p+m,q+n)$$

For the second term:

$$\delta_2 = \sum_{x,y} \lambda_{xy} \left\{ \frac{1}{(2N_c+1)(2N_l+1)} \sum_{k,l|k+xS_l=p,l+yS_c=q} H(xS_l+k,yS_c+l) \right\}$$

For the last term

$$\delta_3 = 2r \sum_{x,y} \frac{1}{(2N_c+1)(2N_l+1)} \sum_{k,l|k+xS_l=p,l+yS_c=q} E(x,y)H(xS_l+k,yS_c+l)$$

A simple gradient can be applied on the correction factors:

$$c^{k+1} = c^k - \alpha_c(\delta_1^k + \delta_2^k + \delta_3^k)$$

and on the Lagrange parameters:

$$\lambda_{xy}^{k+1} = \lambda_{xy}^k + \alpha_\lambda E(x,y)$$

This method has some drawbacks in term of stability and scalability.

The gradients δ_2, δ_3 strongly depend on the number of terms in the sum. This number can be different and introduce strange behaviour in some areas during iterations. These areas depend on the high-to-low window size. The correction factors values of these areas are converging slower. It is always a good feature to have for every variable the same level of error and speed of convergence. When the iteration process is stopped, the error is uniformly distributed. To avoid this problem, δ_2 and δ_3 will be divided by the number of terms in their sum.

3.3.3 Stop tests

The stop test usually done is based on the updating value. What is important here is not the error on the correction factor but the constraints validity and the smoothing factor. The chosen tests are:

$$MAX \left| L(x,y) - \frac{1}{(2N_c+1)(2N_l+1)} \sum_{k=-N_l,..,N_l,l=-N_c,..,N_c} c(xS_l+k,yS_c+l)H(xS_l+k,yS_c+l) \right| < \epsilon_L$$

$$\sqrt{\frac{1}{\#c}\sum_{i,j}[C(i,j)]^2} < \epsilon_c$$

3.3.4 Initialisation

Starting from closer values to the solution improves dramatically the speed of convergence. A good estimate can be obtained using the rough estimation:

$$c(i,j) = c(xS_l,yS_c) = \frac{L(x,y)}{\frac{1}{(2N_c+1)(2N_l+1)} \sum_{k=-N_l,..,N_l,l=-N_c,..,N_c} H(xS_l+k,yS_c+l)}$$

and bilinear interpolation for the other correction factors.

3.4 Testing

Included in annex B.

3.5 Conclusion

Lagrangian approach succeeded to solve a simplified Resolution Enhancement problem, applied on simulated, as on Meteosat 7 images, using acceptable ressource. The extension to real RE problem seems feasible.

Three features allowing to improve the algorithm are pointed out : First, it seems essential to use initialisation. Second, it seems not useful to use the Lagrange parameters. Third, to lower computing time, only pixels that are not verifying the end tests should be updated. For the test on METEOSAT image, only 619 pixels (in low resolution) are not verifying the tests after initialisation. So about only 2% of the data should be updated since the processing is local. A good method should be to make some iterations on bad pixels and then one iteration on all the pixels and so on.

These conclusions must be verified on real configuration problem (and applied on real SEVIRI images).

4 IMPLEMENTATION ON REAL PROBLEM

4.1 Introduction

Let's summarize : The aim of the process main part is to find $cS(i,j)$, the high (or SEVIRI) resolution correction factor, correction to the high resolution SEVIRI based flux estimates $\widetilde{F}_{HR}(i,j)$, so that the down sampling of the corrected flux estimates does reproduce the low resolution GERB L20-product flux $F_{LR}^{L2}(x,y)$.

Comparing to the simplified approach, we will now add the time dimension and take into account of the geolocation difference.

Since the acquisition time frequency is not the same for the two instruments, the Resolution Enhancement processing is divided in two parts. First, we will work at GERB acquisition time t^g. Secondly, we will working out a weighted average to get high resolution flux at SEVIRI acquisition time t^S.

As explained in the last section, the first part of the Resolution Enhancement problem, the search for the $cS(i,j,t^g)$, or the high SEVIRI resolution correction factor at GERB time, can be seen as a problem of minimization of function under 'm' constraints.

4.2 Problem description

As explained in the simplified approach, the main part of the Resolution Enhancement problem, the search for the $cS(i,j,t^g)$, the SEVIRI or high resolution correction factor at GERB time, can be seen as a problem of function minimization under 'm' constraints. The function to minimise is an estimation of the roughness of the correction factors at high resolution cS(i,j). And the 'm' constraints are the equalities between, the down sampling of the high resolution corrected flux estimates, and the low resolution GERB L20-product fluxes.

Let's denote :

1. (x,y) representing a GERB pixel,
2. (i,j) representing a SEVIRI pixel,
3. $F_{LR}^{L2}(x,y)$the low resolution GERB L20-product flux at GERB pixel (x,y),
4. (xS,yS)representing the SEVIRI pixel which geolocation is the closest from the GERB pixel (x,y) geolocation (Indeed, any GERB pixel geolocation is assuming to match with any SEVIRI pixel geolocation.

5. $P^d(i,j)$ the Point Spread Function (PSF)at pixel (i,j) for detector d.
6. $cS(i,j)$ the high (or SEVIRI)resolution Correction factor at SEVIRI geolocated pixel (i,j).
7. m,n=1->N sweeping the PSF area ,
8. $\widetilde{F}_{HR}(i,j)$ the high resolution SEVIRI based flux estimates for pixel (i,j)

Following this notation, the correlation or equality constraint at GERB pixel (x,y) is written :

$$F_{LR}^{L2}(x,y) = \sum_{m,n=1}^{N} \Big[P^{d=y}(m,n)\, cS(xS+m, yS+n) \widetilde{F}_{HR}(xS+m, yS+n) \Big]$$

Since this relation should be checked at each GERB pixel (x,y), we have a number of constraints equal to the number of GERB pixel : m = GERB_IMAGE_WIDTHx GERB_IMAGE_HEIGHT.

The function to minimize, the smoothing estimation function compares each correction factor at high resolution cS(i,j) to the mean of its 8 neighbours :

$$function[cS(i,j)] = \frac{1}{2} \sum_{i,j} \left[cS(i,j) - \frac{1}{8} \sum_{k,l=-1,1} (cS(i+k, j+l)) \right]$$

Minimizing this function is equivalent to impose a linear variation to the correction factors cS(i,j).

4.3 Generalized Lagrange function

The problem seen as a problem of minimization of function under 'm' constraints is then numerically solved using a simple iterative method of minimization of Generalized Lagrange Function.

For the considered real problem, the Generalized Lagrange Function is :

$$\Lambda(cS, r) = \frac{1}{2} \sum_{i,j} \left(Cmean(i,j)^2 \right) + r \sum_{x,y} \left(E(x,y)^2 \right)$$

with
$Cmean(i,j) = cS(i,j) - \frac{1}{8} \sum_{k,l=-1,1} (cS(i+k, j+l))$
$E(x,y) = \sum_{m,n=1}^{N} \Big[P^{d=y}(m,n)\, cS(xS+m, yS+n) \widetilde{F}_{HR}(xS+m, yS+n) \Big] - F_{LR}^{L2}(x,y)$

4.4 Minimization iterative scheme

The method chosen to minimise the Generalized Lagrange Function is an iterative process based on the gradient calculation :

1. Let's start with a first guess of the solution $cS^{(k=0)}$
2. Calculate the Lagrangian gradient (at step k) : $\delta\Lambda^{(k)}(i,j) = \frac{\partial \Lambda}{\partial cS^{(k)}(i,j)}$
3. Calculate next cS estimation : $cS^{(k+1)}(i,j) = cS^{(k)}(i,j) - \alpha\, \delta\Lambda^{(k)}(i,j)$
4. Go back to step 2 (k=k+1).

The choice of the step factor α is of major importance for the convergence speed of this process, and will be discuss in a following section.

Let's develop the gradient of the Lagrange function ($\delta\Lambda(i,j)$), relative to the the correction factor at pixel (i,j).

For the first term, we get :

$\delta_1 = \frac{\delta}{\delta cS(i,j)} \sum_{i,j} \left(Cmean(i,j)^2\right)$
$= Cmean(i,j) - \frac{1}{8} \sum_{m,n=-1,1} \left(Cmean(i+m, j+n)\right)$
And for the second term, we have :
$\delta_2 = \frac{\delta}{\delta cS(i,j)} \sum_{x,y} \left(E(x,y)^2\right)$
$= 2 \sum_{x,y} \left(E(x,y).\frac{\delta E(x,y)}{\delta cS(i,j)}\right)$
$= 2 \sum_{x,y} \left(E(x,y).\widetilde{F}_{HR}(i,j).P^{d=x}(i - xS, j - yS\right)$
with x, y such as the corresponding xS is included in {i-N/2,i+N/2} and yS is included in {j-N/2,j+N/2}

These developments allow us to write the Lagrangian gradient as a function of Cmean and E :

$$cS^{(k+1)}(i,j) = cS^{(k)}(i,j) - \alpha \left(\delta_1(Cmean^{(k)}) + \frac{r}{nb}\delta_2(E^{(k)})\right)$$

- where nb is the number of term in the sum $\sum_{x,y}$ building up δ_2. Indeed, we saw in the simplified approach, gradient $\delta_2$strongly depends of the number of term in the sum. And to avoid scalability problem, it's important to renormalize this term.
- and where r is a ponderation factor aiming to balance importance between the 2 gradient terms $\delta_1$and δ_2. The choice of its value is explained in a following section (see 4.8).

4.5 Initialisation

As iterative scheme, this process requires the choice of good initial values $cS^{(0)}$. The closer to the solution is this first solution guess, the faster is the convergence.

A good first estimation of $cS^{(0)}(i,j)$ value can be derived from the correction factor at GERB resolution $CG(x,y)$ computed in the GERB Processing, applying a bilinear interpolation.

$cS^{(0)}(i,j)$ =Bilinear interpolation of the 4th $CG(x,y)$surrounding pixel (i,j)
with $CG(x,y)$ the low (or GERB)resolution Correction factor at GERB geolocated pixel (x,y).

4.6 Stop tests

The iterative process is stopped, meaning the updating value $cS^{(k+1)}(i,j)$ is estimated close enough from one solution, when

the 2 following criteria are valid :

1. Test on the smoothing factor :

$$\epsilon_S = \sqrt{\frac{\sum_{i,j}\left(Cmean(i,j)^2\right)}{nb(Cmean)}} < \epsilon_S^{required}$$

2. Test the constraint validity :

$$\epsilon_E = Max_{x,y}\left|E(x,y)\right| < \epsilon_E^{required}$$

In the current processing version, the following values have been chosen :

$\epsilon_S^{required}$	0.001
$\epsilon_E^{required}$	$0.01 * Max(\widetilde{F}_{HR})/2$

4.7 Selection of parameter α

The α parameter defines the scale of the correction step and therefore is of preeminent concern for the success of the iterative process.

Four methods to selectionnate α where implemented and compared :

4.7.1 Method of constant step

The first one, the simplest, consist to choose α constant for all steps. The α value is fixed at the first iteration such as corrections are all inferior to 1.5 %. More precisely, we evaluate the maximal value of the correction : maxcorr $= Max_{i,j} \left|\delta\Lambda^{(1)}(i,j)\right|$, then we take $\alpha = \frac{0.015}{maxcorr}$. This methods ensure stability of the iterative process but, as drawback, can be very slow.

In order to speed up the convergence, we tried other methods where αvalue is reevaluated at each iteration. The common principle is to chose among different αvalue the one that minimises a good estimation of combined errors, without losing too much computing time.

4.7.2 Estimation of combined error

- Analytical calculation of ϵ_S

Different values of the smoothing factor ϵ_S, according to different αvalues, can be easily and quickly calculated at the same step k, thank to the following development :

$$\epsilon_S^{(k)} = \sqrt{\frac{\sum_{i,j}\left(Cmean^{(k)}(i,j)^2\right)}{nb(Cmean)}} = \sqrt{\frac{\sum_{i,j}\left(cS^{(k)}(i,j) - \frac{1}{nb(C)}\sum_{k,l} cS^{(k)}(i,j)\right)^2}{nb(Cmean)}}$$

$$\epsilon_S^{(k+1)} = \sqrt{\frac{\sum_{i,j}\left(cS^{(k)} - \alpha\,\delta\Lambda^{(k)} - \frac{1}{nb(C)}\sum_{k,l}\left(cS^{(k)} - \alpha\,\delta\Lambda^{(k)}\right)\right)^2}{nb(Cmean)}}$$

$$\epsilon_S^{(k+1)} = \sqrt{\frac{\sum_{i,j}\left(Cmean^{(k)} - \alpha\,\delta\Lambda mean^{(k)}\right)^2}{nb(Cmean)}} \text{where } \delta\Lambda mean = \delta\Lambda - \frac{1}{nb}\sum_{k,l}\delta\Lambda$$

If we precalculate the 2 coefficients $c_1 = \frac{\sum_{i,j}\left(Cmean^{(k)}\,\delta\Lambda^{(k)}\right)}{nb(Cmean)}$ and $c_2 = \frac{\sum_{i,j}\left(\delta\Lambda^{(k)}\right)^2}{nb(Cmean)}$, $\epsilon_S^{(k+1)}$ can be evaluated for several αvalues without waste of computing time :

$$\epsilon_S^{(k+1)} = \sqrt{\epsilon_S^{(k)} - 2c_1\,\alpha + c_2\,\alpha^2}$$

- Local evaluation of the correlation constraint ϵ_E

Since time to fully compute the correlation constraint ϵ_E is very expensive, correlation constraint is evaluated, for different αvalues, locally (We note ϵ_E^*). Meaning the influence of this different αvalues, on the correlation constraint, is not checked at all the GERB pixel (x,y), but only at M GERB point. These M GERB points are selected at the next iteration, there are those where the correlation relation was the worst. (e.i. pixel (x,y) where $|E(x,y)|$is maximum)

Finally, in order to get one estimation of error, these two errors factors ($\epsilon_E^*$and ϵ_S) are combined with the ponderation factor r : $\epsilon^*_{combined}(\alpha) = \epsilon_S(\alpha) + r\,\epsilon_E^*(\alpha)$. Among all the tested αvalues the one minimizes $\epsilon^*_{combined}(\alpha)$is selected and then applied to the full image.

Based on this combined error estimation, 3 different methods were tested, different because of the choice of the tested αvalues.

4.7.3 Method of the N regularly spaced values

This methods consist to make a selection among N αvalues taken regularly spaced in the range [0,α_{MAX}], with $\alpha^{(k)}_{MAX} = \frac{0.05}{maxcorr(k)}$.

More precisely, $\epsilon^*_{combined}(\alpha)$ is computed for N αvalues : $\frac{1}{N}\alpha_{MAX}$, $\frac{2}{N}\alpha_{MAX}$, ..., $\frac{N-1}{N}\alpha_{MAX}, \alpha_{MAX}$. Then, the αvalue minimizing $\epsilon^*_{combined}(\alpha)$is kept out.

4.7.4 Recursive bracketing method

As an improvement of the last method, here we also search for the best α value in a given bracket [0,α_{MAX}], but using an iterative process we get, for the same computing time, i.e. the same number of $\epsilon^*_{combined}(\alpha)$computation, a more precise result. This methods consist to divide by 2, at each step, the size of the bracket assuming to include the best α. The subbracket selection requires at each step 2 calculations of $\epsilon^*_{combined}(\alpha)$. The iterative process is stopped, either after a fixed number of iteration, or when the size of the bracket reaches the required precision.

4.7.5 Golden Section Search

Golden section search is an iterative method to quickly isolate the minimum of a function (for instance f(α)) in a given bracketing triplet of abscissas ($\alpha 1, \alpha 2, \alpha 3$) (ref. [2] : Numerical recipe in C). At each step, the routine reduces the bracket size until the bracket reaches the required precision.

From an external point of view, this minimization method looks more efficient than the last one. But, for our specific case of minimization , several tests show us the opposite. The first problem in our case is that we don't minimise the real function, the full error calculation $\epsilon_{combined} = \epsilon_S + \epsilon_E$, but a local estimation of the error function $\epsilon^*_{combined}$. Since this minimum search is not rigorous, we have to limit the first bracket, we take [0,α_{MAX}]. The second problem of this method is that its efficiency is based on an initial bracket really including a minimum, therefore, when choosing the first bracket as [0,α_{MAX}], we are never sure we bracket a minimum of $\epsilon^*_{combined}(\alpha)$.

4.7.6 Conclusion

After several tests on simulated, and on real Eumetsat images, we chose the recursive bracketing method as the more efficient regarding to computing time savings.

4.8 Ponderation factor r

The factor r is aiming to balance the importance between the 2 terms of the Generalized Lagrange function :

$$\Lambda(cS, r) = \frac{1}{2} \sum_{i,j} \left(Cmean(i,j)^2 \right) + r \sum_{x,y} \left(E(x,y)^2 \right)$$

The choice of a well fitted ponderation factor r is of major importance for the convergence of our iterative process[1]. On one hand, if r is chosen too big the problem becomes wrong conditionned and generate numerical difficulties. On the other hand, r must be chosen big enough to reach an optimal solution (e.i. to reach low enough value of E).

We can also see the balance role of the factor r at an other level. The factor r is ponderating the 2 parts of the correction applied to cS, through the Lagragian gradient :

$$cS^{(k+1)}(i,j) = cS^{(k)}(i,j) - \alpha \left(\delta_1(i,j) + \frac{r}{nb} \delta_2(i,j) \right)$$

The first part (δ_1)is corresponding to the smoothing constraint, and the second part(δ_2)is corresponding to the correlation constraint.

Intuitively, it seems good to ponderate the 2 parts of the correction, at each step, proportionally to ratio between the 2 errors factors at this step.

With others words, if the correlation criterion (ϵ_E) is less satisfied than the smoothing criterion (ϵ_S) , the correction part relative to the correlation constraint(δ_2) must be bigger than the first part(δ_1), in the same rate.

So, we should chose r such as we get grossly at each point (i,j) :

$r \frac{\delta_2}{\delta_1} = \frac{\epsilon_E^{rel}}{\epsilon_S^{rel}}$ with $\epsilon_E^{rel} = \frac{\epsilon_E}{\epsilon_E^{required}}$ and $\epsilon_S^{rel} = \frac{\epsilon_S}{\epsilon_S^{required}}$

Several tests on simulated data and on Eumetsat images [see : Developping and validation tests in annex C] show us to chose :

$$r = \frac{\frac{\overline{|E^*|}}{\epsilon_E^{required}}}{\frac{\epsilon_S}{\epsilon_S^{required}}} \frac{\|\delta_1\|^2}{\overline{|\delta_2|}}$$

where :

- $\|\delta_1\|^2 = \sqrt{\frac{\sum_{i,j} (\delta_1(i,j)^2)}{nb(\delta_1)}}$

- $\overline{|\delta_2|} = \frac{\sum_{i,j}|\delta_2|}{nb(\delta_2)}$
- $\overline{|E^*|} = \frac{\sum_{x,y}|E^*|}{nb(E^*)}$ with x, y such as $|E(x,y)| > \epsilon_E^{required}$

4.9 Convergence improvement attempts

In order to reduce the global computing time of the RE process, we tried, through different ways, to reduce, either the number of calculation at each iteration, or the total number of iteration, meaning speed up the convergence.

4.9.1 Method of the conjugated gradient (or Flechter and Reeves method)

In stand to correct the factors cS at the step k in the direction of the Lagrangian Gradient at this step $\delta\Lambda^{(k)}$,we take into account of the gradient direction at the step k-1, and we combined both directions. The algorithm described below become :

1. Start with a first guess of the solution $cS^{(0)}$and $D^{(0)} = \delta\Lambda^{(0)} = \frac{\partial\Lambda}{\partial cS^{(0)}}$
2. Calculate next cS estimation : $cS^{(k+1)}(i,j) = cS^{(k)}(i,j) - \alpha\, D^{(k)}$
3. Calculate next Lagrangian gradient : $\delta\Lambda^{(k+1)}$and the next conjugated gradient $D^{(k+1)} = \delta\Lambda^{(k+1)} - \frac{\left\|\delta\Lambda^{(k+1)}\right\|^2}{\left\|\delta\Lambda^{(k)}\right\|^2}\delta\Lambda^{(k)}$
4. Go back to step 2.

This method sometimes allow us to reach a convergence speed faster than with simple gradient algorithm. The required critera are verified after less iteration. As drawback, the conjugated gradient calculation requires a bit more storage and a little bit more computing time at each iteration. For the tested cases, the increase of RAM storage (1 high resolution image) is acceptable. But the computing time increase is higher than the time savings with less iteration. So we decided to not keep this method.

4.9.2 Limitation of the correction area

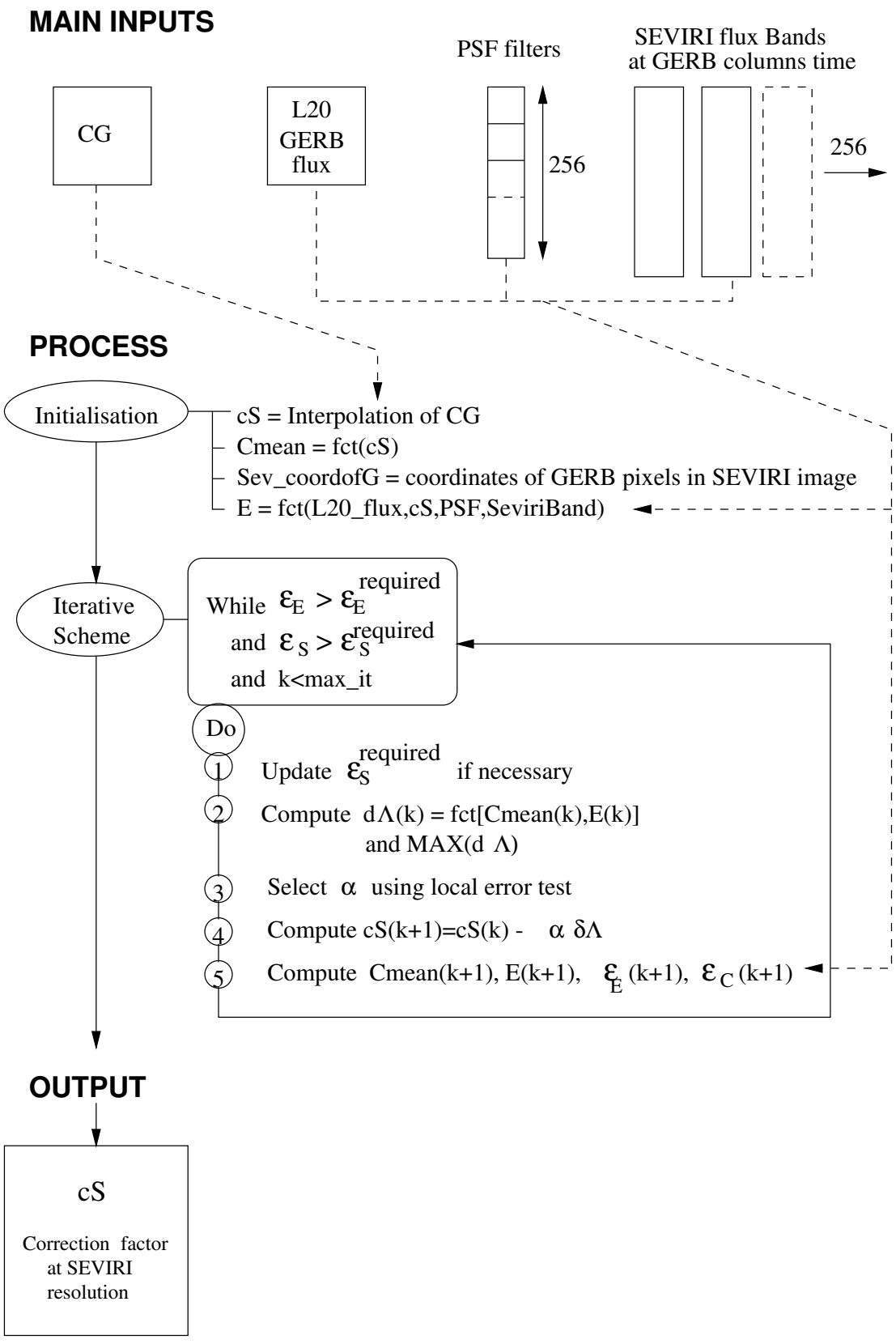
An other idea to save computing time is to limit the number of points (i,j) where cS(i,j) is updated. By this way we reduce the number of point where $\delta\Lambda(i,j)$ is computed, one of the more time-expensive calculation.

During the first iterations, the factor cS is updated at all the SEVIRI pixel (i,j), and so, $\delta\Lambda^{(k)}$must be calculated at all the SEVIRI pixel. Then, when the number of points where the correlation criterion is not satisfied, is inferior to a specified value (for example 50), the factor cS and the gradient $\delta\Lambda$ are updated only at the pixels (i,j) located in the neighbourhood of this bad ϵ_E values (i.e. in PSF area where $\epsilon_E(x,y) < \epsilon_E^{required}$). So during the last iterations, we save a big part of the time used to compute $\delta\Lambda$.

4.9.3 Updating $\epsilon_S^{required}$

Despite the limitation of the correction area during the last iterations, computing time remains too long for some configuration of tested images. In these case, we save computing time by lowering our smoothing requirement. After N (for example N=15) iterations interval, the smoothing error required $\epsilon_S^{required}$ is incremented.

4.10 Algorithm scheme



For more detail about program structure, see Annex A.

4.11 Testing

Included in annex C.

5 Resolution Enhancement averaged at SEVIRI time

The first part of the Resolution Enhancement system, described in section4, gives as output the high (or SEVIRI) resolution correction factor at GERB time : $cS(i,j,t^g)$. In the second part we will average these estimations at GERB time to SEVIRI time, and then compute the final output, the L2 high resolution corrected flux averaged at SEVIRI time $F^{L2}_{HR}(i,j,t^s)$.

5.1 Correction ratio time averaging

At each SEVIRI pixel (i,j), a weighted average on 15 minutes (according to SEVIRI image acquisition period) is worked out.

$$cS(i,j,\overline{t^s_{ij}}) = \sum_r \left[I(\overline{t^g_{ijr}})\, cS(i,j,t^g_{ijr}, g^s_{ij}) \right]$$

With :

$\overline{t^s_{ij}}$ means time average on 15 minutes surrounding the acquisition time of SEVIRI pixel (i,j) at slot r.

$\overline{t^g_{ij}}$ represents a time period of 5min14 surrounding the acquisition time according to SEVIRI pixel (i,j) at GERB slot r.

The weighting factor $I(\overline{t^g_{ijr}})$is taken proportional to overlap factor of time period $\overline{t^g_{ijr}}$on time period $\overline{t^s_{ij}}$.

5.2 Multiplication to get level 2 high resolution flux

Finally, the level 2 high resolution flux products can be calculated at each SEVIRI pixel (i,j) as a multiplication :

$$F^{L2}_{HR}(i,j,t^s) = cS(i,j,t^s)\, \widetilde{F}_{HR}(i,j,t^s)$$

It's maybe important to notice that only the correction factor is averaged, and then multiplied to the high resolution SEVIRI based flux estimation at SEVIRI acquisition time t_S. So the resulting corrected high resolution flux is a 'snapshot' flux instead of a real average flux.

This final output is stored in directory L20_SEVIRI/L2S_SF (or L20_SEVIRI/L2S_TF)with a file name following the convention

L2S_SF_<date>_<version>.rma (or L20_TF_<date>_<version>.rma).

References

[1] Michel Minoux, "Programmation mathematique",Dunod,1983.

[2] Press, Teukolsky, Vetterling, Flannery, "Numerical recipes in C", Cambridge University Press, v.2 1992.

[3] S.Dewitte, "RGP: Overview ", RMIB technical report, MSG-RMIB-GE-TN-4,1999.

[4] L.Gonzalez, A.Hermans, "RGP: GERB Processing", RMIB technical report, MSG-RMIB-GE-TN-6,1999.

[5] N.Clerbaux, S.Dewitte, "RGP-SP: Spectral Modelling ", RMIB technical report, MSG-RMIB-GE-TN-5,1999.

[6] N.Clerbaux, S.Dewitte, "RGP-SP: Scene Identification", RMIB technical report, MSG-RMIB-GE-TN-7,1999.

[7] N.Clerbaux, S.Dewitte, "RGP-SP: Angular Modelling ", RMIB technical report, MSG-RMIB-GE-TN-8,1999.

A Program structure description

see file : RE_progdescription.lyx (in directory: /rain/aline/Docu/gerb/Resol/)

B Testing on simplified approach

see file : RE_test_simple.lyx (in directory: /rain/aline/Docu/gerb/Resol/simple_approach)

C Testing on real problem

see file : RE_test.lyx (in directory: /rain/aline/Docu/gerb/Resol/real_problem)