3. BRDF Upscaling

The BRDF up-scaling algorithm uses “energy conservation” as resampling method (see ATBD, page??). The developed tool aggregate BRDF 1km-tiles (with MODIS-SIN projection) to lat/lon grids which can be with different resolutions (0.5°x0.5°, 0.05°x0.05°,etc) .

The process of up-scaling can be divided into two major sequential stages. First stage creates a set of files for linking each resolution cell in 1km-tile to its corresponding resolution cells in destination grid. And then the second stage is about aggregation and creation up-scaled products. The first stage can be executed once and save its resulted files. So, only the second stage will be launched every time to create up-scaled product. This allows us to avoid re-computing destination resolution cells, which is considered as a time consuming process.

3.1 1st Stage: create linking files

Let $cell_t(i)$ be ith resolution cell in source product I_t (with t refers to tile), and $cell_{rp}(j)$ be j^{th} resolution cell in p^{th} part of destination (up-scaled) product I_{dp}, and $cell_{trp}(i)$ be the projected cell of $cell_t(i)$ into I_{dp}. The forms of $cell_t(i)$ and $cell_{rp}(j)$ are either square or rectangle with equal spatial resolution, however the form of projected cell $cell_{trp}(i)$ can take polygon geometry and be with different spatial resolutions, especially in high latitude.

The up-scaled product I_d could be divided into many equal and non-overlapped parts I_{rp} (with p $=1,2, 3\ldots$ number of parts) as the up-scaling by energy conserving requires a big amount of RAM memory, especially in the case of high resolution destination grids (0.05° and less). For 0.05°, I_d is often divided by 16 parts (Latitude-axis by 4 and Longitude-axis by 4). In the final step of upscaling process all partitions I_{dp} will be re-gathered to form a single product I_d.

The common area $cell_t(i)$ and $cell_{rp}(j)$ between ratio is defined as follows:

$$car_{trp}(i,j) = area(cell_{trp}(i) \cap cell_{rp}(j))/min(area(cell_{trp}(i)), area(cell_{rp}(j)))$$

If $car_{trp}(i,j)>0$, the tool measures the distance between their centroids. This measurement is useful for determining which $cell_{trp}(.)$ is closest to $cell_{rp}(.)$. Because some bands like “Wighted_Number_Of_Observations” is up-scaled by using “Nearest Neighbor method” as resampling method which can’t be done by common area ration measurement (see figure ???).

Indeed, the tool allow user to select for each band its appropriate resampling method between the three available methods: Energy-Conserving, Nearest-Neighbor and Majority Filter. It is recommended to replace Energy-Conserving by Nearest-Neighbor if destination resolution is equal or higher than 0.01°x0.01° as the area of grid cells is closer to the ones of 1km-cell in equator.

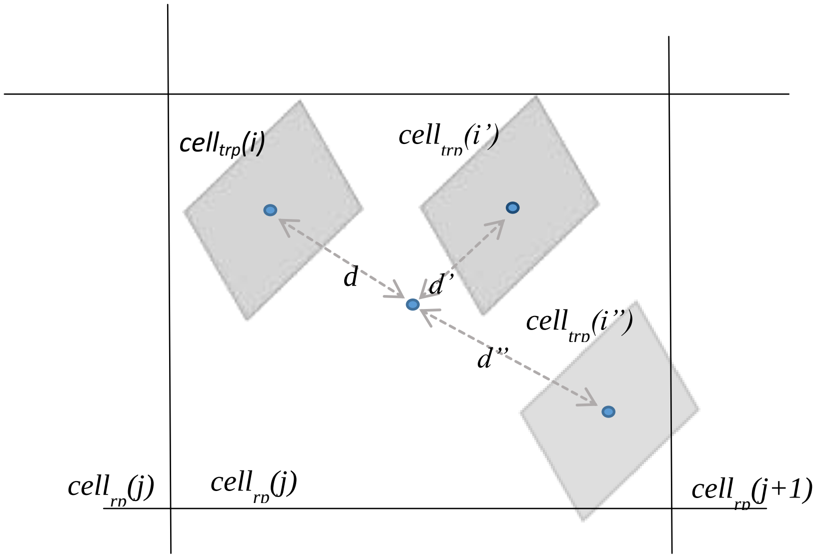


Figure 4: In addition to ratio of common area the distance between will be saved to determine the closest cell.

The output of 1st stage is a set of text files so that each file links the cells of a 1km-tille-modis-sin grid to a part of the destination grid. The name of output file is composed like that:

t.r.p.txt (or tile.resolution.part.txt)

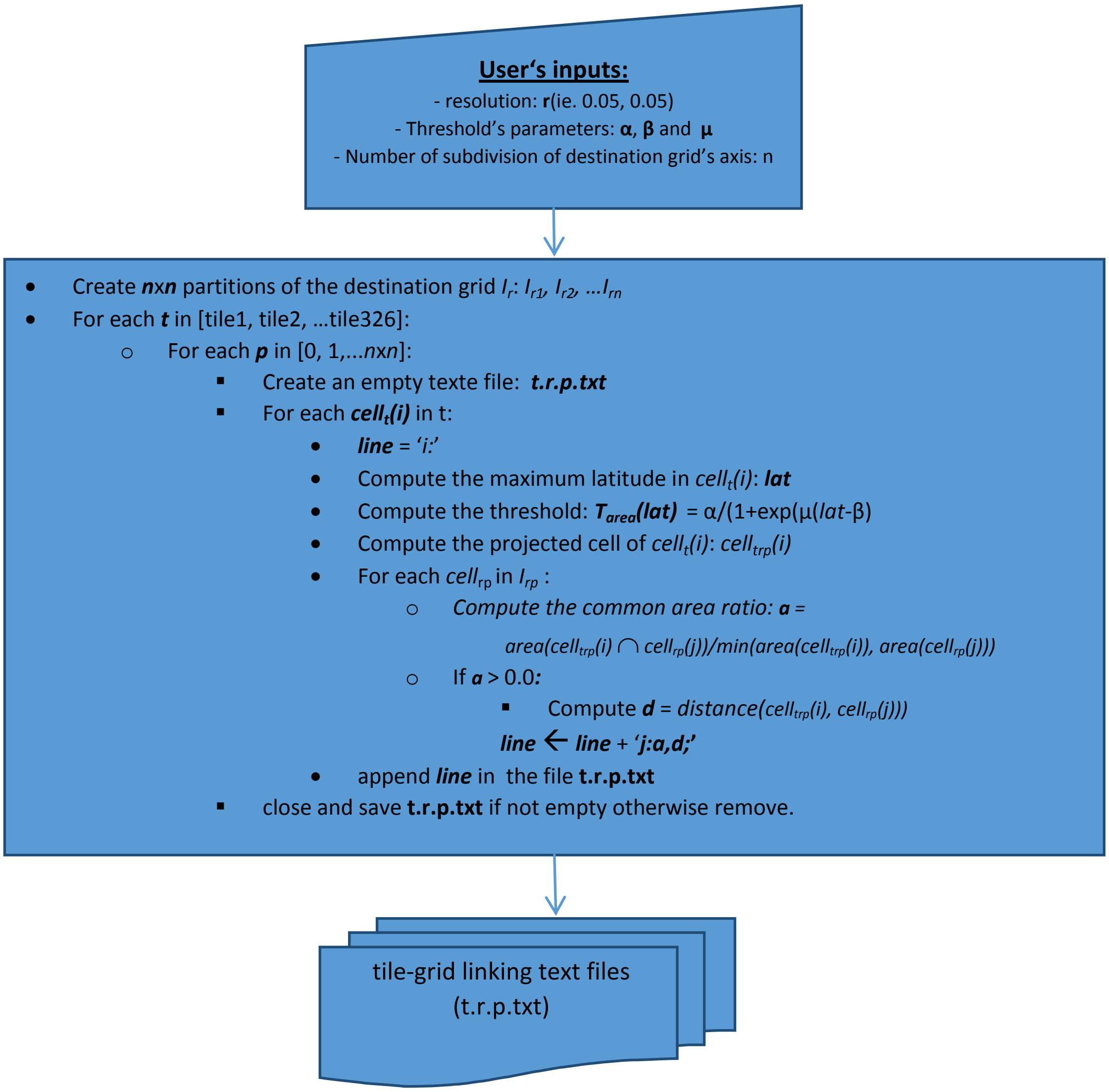
With "t" refers to the name of a 1km-tile-modis-sin (ie. h17v03); "r" refers the spatial resolution of up-scaled product (*d*='005' for the grid 0.05◦x0.05◦ and *d*='05' for the grid 0.5◦x0.5◦) and "*p*" refers to the p^{th} part of the destination of the destination grid (*p* = 0,1,2…15 in case of 4x4 partitions).

A line in "t.d.p.txt" files takes this form:

i = j1:a1,d1; j2:a2,d2; j3:a3,d3 …

With *i* refers to a the $cell_{t}(i)$, *j1* refers to $cell_{rp}(j1)$, as refers to the ratio of common area between $cell_{trp}(i)$, $cell_{rp}(j1)$ and d1 refers to distance between the centroids of $cell_{trp}(i)$ and $cell_{rp}(j1)$. We can say the same for *j2, a2, d2, etc..*

The figure??? shows the flowchart of the linking process .



User's inputs:

- resolution: **r**(ie. 0.05, 0.05)
- Threshold's parameters: **α**, **β** and **μ**
- Number of subdivision of destination grid's axis: n

```
• Create nxn partitions of the destination grid $I_r$: $I_{r1}, I_{r2}, \ldots I_{rn}$
• For each t in [tile1, tile2, …tile326]:
    ○ For each p in [0, 1,…nxn]:
        ▪ Create an empty texte file: t.r.p.txt
        ▪ For each $cell_t(i)$ in t:
            • line = 'i:'
            • Compute the maximum latitude in $cell_t(i)$: lat
            • Compute the threshold: $T_{area}(lat) = \alpha/(1+\exp(\mu(lat-\beta)$
            • Compute the projected cell of $cell_t(i)$: $cell_{trp}(i)$
            • For each $cell_{rp}$ in $I_{rp}$ :
                ○ Compute the common area ratio: a =
                      $area(cell_{trp}(i) \cap cell_{rp}(j))/min(area(cell_{trp}(i)), area(cell_{rp}(j)))$
                ○ If a > 0.0:
                    ▪ Compute d = $distance(cell_{trp}(i), cell_{rp}(j)))$
                  line ← line + 'j:a,d;'
            • append line in the file t.r.p.txt
        ▪ close and save t.r.p.txt if not empty otherwise remove.
```

tile-grid linking text files
(t.r.p.txt)

Figure 5. Algorithm of the first stage of up-scaling process (linking).

2nd Stage: Upscalling

In this stage the aggregation is being done. So, for each part of destination grid I_{rd} the system checks for each source product I_t if the linking file "t.r.d.txt" exits, and then uses this txt file to aggregate directly the values from source product's cells to the destination product's if the condition on "common area ratio" is satisfied.

We admit that $cell_t(i)$ can be liked to $cell_{rp}(j)$ only if :

$$car_{trp}(i,j) \geq T_{area}$$

With T_{area} be a threshold value on the ration of common area between $cell_{trp}(i))$ and $cell_{rp}(j))$. So, the link will be established only if there is a complete containing or the intersection area is relatively significant.

The threshold T_{area} varies with latitude as the area of the projected $cell_{trp}(.)$ becomes small and its geometry narrow and tall as the latitude approaches to the pol regions (see figure ???) which creates empty cells if we work with a static threshold for all latitude values (see figure ???). The proposed function is implanted as follows:

$$T_{area}\,(lat) = \alpha/1+exp(\mu(|Lat|-\beta))$$

With *lat* denotes the highest latitude of $cell_{rp}(j)$, α=0.6, β=80 and µ=1.0 . However, the three values of α, β and µ can be parameterized.

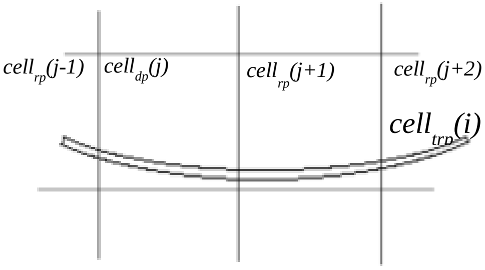


Figure 6. An example of a projected 1km-modis-sin cell into lat/lon grid in high latitude.

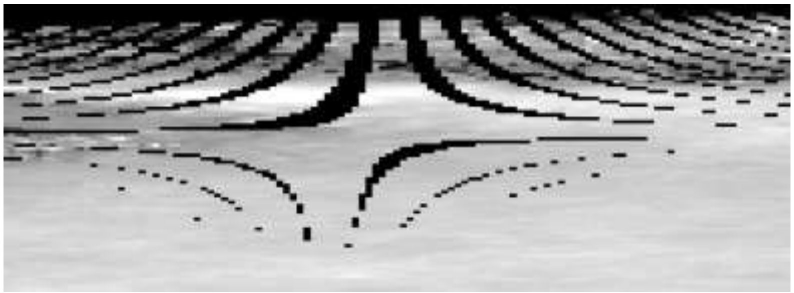


Figure 7. Empty cells are resulted in high latitude of up-scaled product if we use a static threshold (her $T_{area}\,(lat)=0.6$).

The figure ?? shows the plot T_{area} (lat). With “α=0.6, β=80, µ=1.0” the threshold value falls from ~0.6 to ~0.9 when latitude exceeds +/-80°.

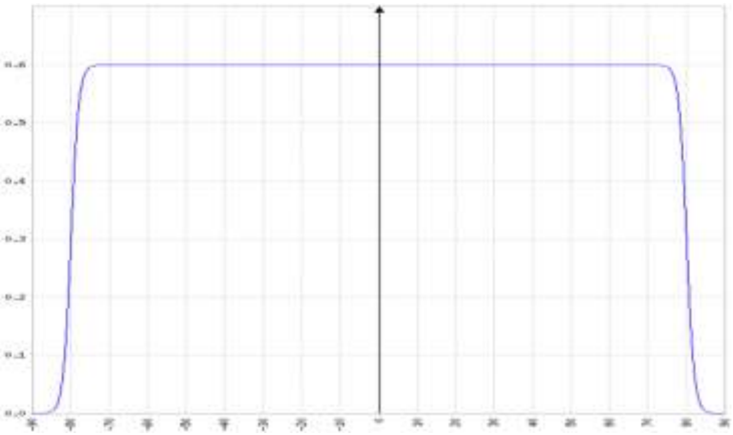


Figure 8. $T_{area}(lat) = 0.6/(1+exp(|lat|-80)$

In high altitude and for high resolution (<0.05◦), the energy conserving method raises an issue of circle marks apparition in up-scaled uncertainty matrix (C). It is due to the significant different of number of samples (number of observations) between adjacent high altitude cells in destination grid (see figure ???).

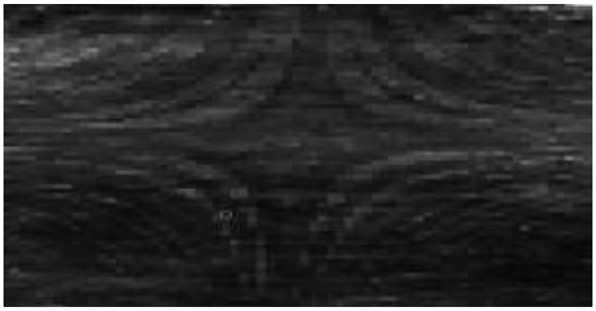


Figure 9. In polarregions, uniform marks appear on C matrix bands up-scaled product 0.05◦.

To fix that issue a function ***numObs(lat)*** has been introduced to limit the number of considered samples regarding the latitude. It is proposed as follows:

$$numObs(lat) = exp(-\tau(|lat|- \beta))+\eta.$$

With *lat* denotes the highest latitude of $cell_{rp}(j)$, η be the minimum of the number of samples that will be selected regardless latitude, denotes the altitude (it is the same β in equation???), τ controls the slope degree. For 0.05◦ the values of these parameters has been fixed as follow: β=80, η=4, and τ=0.1 (see figure???). However, these values can be parameterized and, the function can be disabled, which is recommended for low resolutions (≥0.5◦).

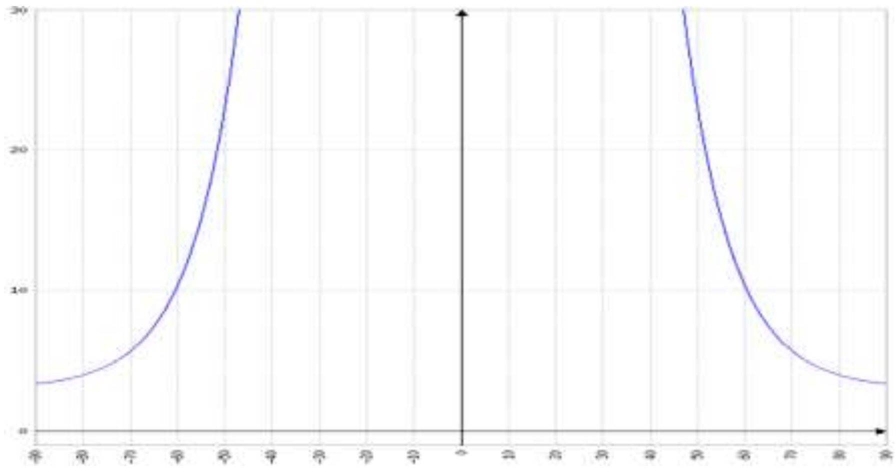


Figure 10. Limit on number of samples regarding the latitude: numObs(lat) = exp(-0.1(|lat|- 80))+4.

The figure ??? show the effects of using a static threshold (ncs)

The figure?? shows the applied algorithm for 2nd Stage.

3.3 Launch

The first stage (matching) can be launched by “/scripts/upscale/match.sh” which takes as input tile name (326 parallel jobs). Other parameters are fixed inside like ouput directory and number of subdivisions, spatial resolution, etc.

The second stage can be launched by “/scripts/upscale/upscale.sh” with doy and year as arguments (644=46*14 jobs). Other paramters is fixed inside that script like ouput of stage1, thresholds, output directory, etc.

4. Albedo for LAI and fAPAR

The process of producing LAI and fAPAR requires as input ten kinds of data:

1) BHR_NIR_Snow: white-sky snow NIR (Near Infra Read wavelength band) albedo;

2) BHR_NIR_NoSnow: white-sky free-snow NIR albedo;

3) BHR_VIS_Snow: white-sky snow VIS (Visible wavelength band) albedo;

4) BHR_VIS_NoSnow: white-sky free-snow VIS albedo;

5) BHR_NIR_Snow_Sigma: Uncertainty of BHR_NIR_Snow;

6) BHR_NIR_NoSnow_Sigma: Uncertainty of BHR_NIR_NoSnow;

7) BHR_VIS_Snow_Sigma: Uncertainty of BHR_VIS_Snow;