**Calculation Processes for NDVI Metrics**

**Version1.1**

**Jiang zhu**

[**jiang@gina.alaska.edu**](mailto:jiang@gina.alaska.edu)

**2011/5/3**

This document describes the NDVI metrics algorithms and running environment of these programs. Programs are realized by IDL codes.

1. Directories and IDL startup file

The programs are resided in jzhu.gina.alaska.edu. The directories used to store different programs or data files are:

1. nps-cesu home directory: CESU\_NDVI=~/nps/cesu/modis\_ndvi\_250m

2. directory for scripts: $CESU\_NDVI/script

unzip\_ndvi.bash, and rename\_ndvi.bash are stored here.

3. work directory: $CESU\_NDVI/wrkdir

store unziped data files,immediate data files, and result data files.

4. idl programs directory: $CENSU\_NDVI/wrkdir/ndvi\_nps

You need set environment variable

Export IDL\_STARTUP=/home/jiang/nps/cesu/startup\_env.pro

In startup\_env.pro, you put one idl command to start envi:

ENVI, /RESTORE\_BASE\_SAVE\_FILES

5. raw zip ndvi data directory: ~/nps/cesu/modis\_ndvi\_250m/TERRA

B. The procedure of the data process:

1. unzip the ndvi files (~nps/cesu/script)

The ndvi files come from EMODIS website. They are zip files. First, download them and store them ~/nps/cesu/modis\_ndvi\_250m/TERRA. Then unzip and rename them by using scripts "unzip\_ndvi.bash" and "rename\_ndvi.bash", respectively.

2. stack and subset 7-day ndvi data, and mark those pixels which are not suitable for NDVI metrics calculation.

inputs: flist\_ndvi, flist\_bq

flist\_ndvi includes 7-day ndvi file names of one year. flist\_bq includes 7-day ndvi\_bq file names of the same year.

program: oneday.pro

subroutines: start\_batch.pro, oneyear\_data\_layer\_subset\_good.pro,

read\_ndvi.pro, subset.pro

outputs: yyyy\_oneyear\_layer\_stack\_0\_200\_snow\_cloud\_n4000. This is stacked, subset, and good data file.

3. stack three years of one-year data into a multiple-year (three years) data

inputs: oneyear-file-list

The oneyear-file-list includes file names of one-year good data files.

program: layer\_stack.pro

subroutines: start\_batch.pro

output: yyyy\_multiyear\_layer\_stack\_0\_200\_snow\_cloud\_n4000. This is a multiple year (normally 3 years) stacked data file.

4. calculate one-year metrics

inputs: a multiple-year data file

program: smooth\_calculate\_metrics\_tile.pro

subroutines: start\_batch.pro, time\_series\_process.pro, interpol\_noextension\_vector.pro, cutoff\_interp.pro, user\_metrics.pro, wls\_smooth.pro

outputs: a metrics data file and a smoothed data file.

C. list of the program files

smooth\_calculate\_metrics\_tile.pro, smooth\_calculate\_metrics.pro, layer\_stack.pro, oneyear.pro, oneyear\_data\_layer\_subset\_good.pro, interpol\_noextension\_vector.pro, read\_ndvi.pro, start\_batch.pro, subset.pro, user\_metrics.pro, cutoff\_interp.pro, wls\_smooth.pro, startup\_env.pro

D. data format description

1. The raw data comes from EMODIS. I use two kinds of files to produce ndvi metrics. One is 250m\_compisit\_ndvi.tif and other is 250m\_composit\_ndvi\_bq.tif. The data range in ndvi.tif is -1999 to 10,000, -2000 is fill value, and the scale factor is 0.0001. Data in ndvi\_bq.tif includes flag information. It indicates the type of the pixels. Types of the pixels are cloud, snow, fog, etc.

2. First, we process one pair of ndvi and ndvi\_bq files by getting rid of those cloud, snow, fog, bad quality, negative reflectance pixels by filling these pixels with another fill value of -4000. Then we convert the data into 0 to 200 byte data by using the formula y=byte(x/100.0+100).In the new data, valid ndvi values are in the range of 100 to 200. The fill values (-2000) are converted into 80, and cloud, snow, fog pixels in raw data are converted into 60.

3. Stack one weekly data into a one year data file. The result file is named as yyyy\_one\_year\_subset\_good.tif (data range is 0 to 200, and data include 42 bands).

4. Combine the year before, the year, and the year after data files into a multiple-year data file. The program layer\_stack.pro finishes this process and produced a 3-year stacked data file named yyyy\_multiyear\_stack\_stack\_0\_200\_set\_snow\_cloud\_n4000.hdr (data range is 0 to 200, and data include 42 bands).

5. Smooth the multiple year stacked data and calculate the metrics of the year. The result data file is named as yyyy\_multiyear\_layer\_stack\_0\_200\_set\_snow\_cloud\_n4000\_smooth.hdr (data range is 0 to 200), and the metrics data is named as yyyy\_multiyear\_layer\_stack\_metrics\_not\_0\_200\_set\_snow\_cloud\_n2000.hdr (data include 12 bands)

E. detailed description of the algorithms

In the whole process of NDVI metrics calculation, three core algorithms are: bad pixel elimination, time-series smooth, and time-series NDVI metric calculation.

1. Bad pixel elimination

"read\_ndvi.bash" does bad pixel elimination.

a. It reads one pair of file (a ndvi file and its related ndvi quality file) into memory. Ndvi data are stored in data1(ns,nl), and ndvi quality flag data are stored in data2(ns,nl).

b. stack the data1 and data2 together to form a two bands data(ns,nl,2). Band1 are data1 and band2 are data2.

c. goes through every pixel of data. First, if pixel is cloud, snow, bad quality, or negative reflectance, set the pixel value (band 1) to -4000, otherwise, keep the pixel value. Second, convert the value of the pixel from range -4000 to 10,000 (integer) into range 0 to 200 (byte) by using equation out=byte(in/100 +100).

d. output the band1 of the data. In the output data, snow, cloud, bad quality, or negative reflectance pixels have byte value of 60b (b presents data type is byte), filling pixels have byte value of 80b, good ndvi pixels have byte values in the range from 100b to 200b.

2. time-series smooth algorithms

"interpol\_nonextension\_vector.pro" and "wls\_smooth.pro" do smooth for time-series vector. The program “interpol\_nonextension\_vector.pro” calls the subroutine named “cutoff\_interp.pro” to get rid of the points which are fill points, replace snow and cloud (with value of 60b) pixels with value 101b or 102b, and interpolate the time-series vector. Wls\_smooth.pro does smooth algorithm. Figure1. Smooth Algorithm Flowchart describes how to do smooth for a time-series vector.

Inputs: ndvi vector, fill\_value(80b), sncld\_fill(60b), and ratio(default =0.6, ratio is used to control if the vector is good for metrics calculation)

get the number of valid points (num) in the ndvi vector and the total number of the vector (num\_point) . Valid points are defined not be equal to the fill\_value.

num/num\_ point >=ratio

get the vector (vv),which is composed of valid points in vector. replace the points in vv which equal to sncld, with random picked value between 101 and 102. interpolate the vector (vv) to get interp\_v, make sure interp\_v has the same length as the original one (ndvi vector).

smooth the interp\_v by weight-least-squre smooth algorithm, original developed by Daniel L. Swets (1999). First, we weight each point with 0.5, then we check each point in interp\_v to adjust the weight. If the value of the point is largest among its two adjective points, we weight the point with 1.5; if the point is the smallest among its two adjective points, we weight the point with 0.005. We calculate the estimate value of each point to guarantee that the estimated values make the sum of weighted squared errors to be the minimum value.

return smoothed vector (smooth\_v )

no

Figure1. Flowchart of Smooth Algorithm

3. NDVI metrics algorithm

“user\_metrics.pro” does metrics calculation for a smoothed vector. It accepts raw vector, smoothed vector, band name vector. It outputs metrics vector. The main steps are: Step a, Check the data in smoothed vector to make sure they are suitable to calculate metrics. If the values of the points are the same, this means the pixel is not suitable for ndvi metrics calculation. Set the flag in output metrics vector, then return. If the index of the maximum value is outside of reasonable range, we also do not calculate the metrics, set flag=0, and return. Step b, after the smoothed\_v passes the tests in step a, we retrieve onset index and end-of-greenness index. First, we retrieve onset index. We calculate three possible indices: idx\_max\_dev, the index of the maximum value in the derivative of smoothed\_v; idx\_max\_dediff, the index of the maximum value in the derivative of the difference between smoothed\_v and backaward moving average (diff); idx\_cross, the index where the values in diff change from negative to positive. Then we decide which index is the onset-idx by comparison of the three indices. We use thre similar method toretrieve the end-of-greenness index. Step c, after we get onset\_idx and end-of-greenness idx, we can easily calculate other metrics items. Figure 2 is the flowchart for NDVI metrics algorithm.

Inputs: interp\_v, smoothed\_v, bandname\_v. output: metrics\_v. metrics\_flg=0 (not valid)

Is the smoothed\_v a filling pixel?

Find the onset point: calculate the backward moving average (bwv) of smoothed\_v, the derivative (dev) of smoothed\_v, and the derivative (dediffv) of diffv=smoothed-v-bwv; find the index (idx\_max\_dev) of the maximum value from cutoff\_bg to idx\_max\_v in dev, the index (idx\_max\_dediffv) of the maximum value from cutoff\_bg to idx\_max\_v in dediffv; get idx\_greater=max(idx\_max\_dev,idx\_max\_dedif), idx\_smaller=min(idx\_max\_dev,idx\_max\_dediff); find the index (idx\_cross) of point in diffv where value change sign from negative to positive; if idx\_cross is not exist, onset\_idx= idx\_greater; if idx\_cross exists, we consider three cases: idx\_cross<idx\_smaller, onset\_idx=idx\_smaller, idx\_smaller <=idx\_cross<=idx\_greater, onset\_idx=idx\_cross; idx\_cross>idx\_grreater, onset\_idx=idx\_greater.

cutoff\_bg and cutoff\_ed gives the reasonable start and end of season index range. Idx\_max\_v is the index of maximum value in smoothed\_v.

cutoff\_bg <=idx\_max\_v<=numofvector-cutoff\_ed

Find the end of greenness point: do the similar process as “find the onset point”, except we use forward moving average, and work on points from idx\_max\_v to numbeofvector-cutoff\_ed.

From onset\_idx and endgreen\_idx, we can convert them into day. Because one idx corresponds to a 7-day period, we pick the mid-day of the 7-day period as onset or end- of-green day. Other merics items are fairly easy to calculate. For NDVI value calculation, we use values in the non-smoothed data vector. The results are stored in metrics\_v, and set metrics\_flag=1 (means valid).

return metrics\_v

yes

Figure 2 NDVI Metrics Algorithm Flowchart