連載講義

Python を利用した衛星画像の大気補正 第 2 回 標準大気モデルを使った衛星画像の大気補正

眞子直弘*

Atmospheric Correction of Satellite Image Using Python, Lecture Series 2: Atmospheric Correction with a Standard Atmosphere Model

Naohiro MANAGO*

1. はじめに

本講義では Python を利用して衛星画像の基礎的な大気補正を行う方法を解説する。第 2 回目の今回は Python を使って放射伝達コード 6SV[1)2)] を制御し，6SV に組み込まれた標準大気モデルを使ってひまわり 8 号衛星画像の大気補正を行う。

2. 大気補正とは

太陽光によって照らされている地表面を衛星から見ると，主に以下の 6 成分からなる放射輝度の総和（L_{obs}）が観測される。（1）L_{ps}：大気中で単散乱された成分（2）L_{pm}：大気中で多重散乱された成分（3）L'_{pm}：対象周辺（Environment，反射率 ρ_e）で反射された後に大気中で単散乱された成分（4）L_{gd}：直接的に対象領域（Target，反射率 ρ）に到達し，反射された成分（5）L_{gi}：大気中で散乱されて対象領域に到達し，反射された成分（6）L'_{gi}：対象周辺で反射された後に大気中で単散乱されて対象領域に到達し，反射された成分。これらはさらに以下の 3 つに分類することができる。（1）L_{atm}：大気中だけで散乱された成分（L_{ps}，L_{pm}）（2）L_{env}：対象周辺で反射された成分（L'_{pm}）（3）L_{tar}：対象領域で反射された成分（L_{gd}，L_{gi}，L'_{gi}）。L_{gd} 以外の成分は大気科学の研究にとって重要な情報を含んでいるが，地表面の研究にとっては余分な情報なので，これを取り除く必要があり，そのことを大気補正と呼んでいる[3)]。

3. 6SV を使った衛星画像の大気補正

3.1 6SV の概要

6SV は人工衛星等で測定される地表面の放射輝度を計算したり，放射輝度値に含まれる大気の影響を見積り，それを取り除くこと（大気補正）が出来る放射伝達シミュレーションコードである。6SV という名前は “Second Simulation of a Satellite Signal in the Solar Spectrum － Vector” に由来している。元々は偏光を考慮しないスカラーコードだったが，2005 年に偏光を考慮したベクターコード（6SV 1.0 Beta）がリリースされた。現在の最新版は 6SV 2.1 である。6SV のソースコードやマニュアルは Web ページからダウンロード出来る[1)]。

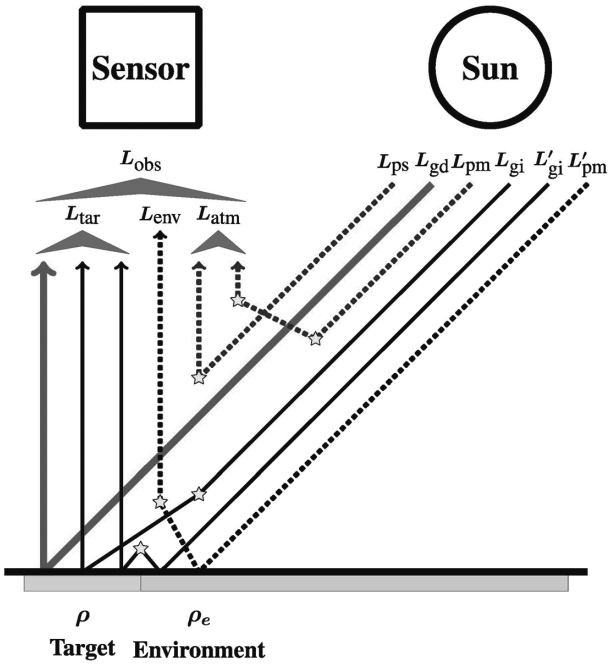

Fig. 1 Radiances observed with a satellite sensor

6SV 内部では前述の L_{atm}，L_{env}，L_{tar} は以下のように計算される[2)]。

（2016. 10. 26 受付）

* 千葉大学環境リモートセンシング研究センター
〒263-8522 千葉市稲毛区弥生町 1-33

* Center for Environmental Remote Sensing, Chiba University

$$L_{\mathrm{atm}}=\frac{1}{\pi}\frac{\int S(\lambda)E_\lambda\mu_s T_g^\lambda(\theta_s,\theta_v)\rho_\lambda^a d\lambda}{\int S(\lambda)d\lambda} \quad (1)$$

$$L_{\mathrm{env}}=\frac{1}{\pi}\frac{S(\lambda)E_\lambda\mu_s T_g^\lambda(\theta_s,\theta_v)\dfrac{T^\lambda(\theta_s)t_d^\lambda(\theta_v)\langle\rho_\lambda\rangle}{1-\langle\rho_\lambda\rangle s_\lambda}d\lambda}{S(\lambda)d\lambda} \quad (2)$$

$$L_{\mathrm{tar}}=\frac{1}{\pi}\frac{S(\lambda)E_\lambda\mu_s T_g^\lambda(\theta_s,\theta_v)\dfrac{T^\lambda(\theta_s)e^{-\tau/\mu_v}\rho_\lambda}{1-\langle\rho_\lambda\rangle s_\lambda}d\lambda}{S(\lambda)d\lambda} \quad (3)$$

ここで，S：フィルター関数，E_λ：大気上端太陽放射照度，τ：大気の光学的厚さ，t_d^λ：拡散透過率，T^λ：透過率（$e^{-\tau/\mu}+t_d$），T_g^λ：気体分子吸収の透過率，s_λ：球面アルベド，ρ_λ：地表面反射率，ρ_λ^a：大気の反射率，λ：波長，θ：天頂角，μ：cosθ である。添え字 s，v はそれぞれ太陽方向および視線方向の量であることを表し，記号 〈〉 は空間平均を表す。6SV の大気補正モードを有効にすると，入力された（大気補正されていない）見かけの反射率 ρ_i^* または見かけの放射輝度 L_{obs} を再現するように大気補正された反射率 ρ_{ac} が求められる。（大気補正モードでは，地表面は一様反射する Lambertian として扱われる。）

$$\rho_i^*=\frac{\pi L_{\mathrm{obs}}}{E\mu_s} \quad (4)$$

$$\rho'_{\mathrm{ac}}=\frac{\rho_i^*-\rho_{\mathrm{atm}}}{T_g(\theta_s,\theta_v)T(\theta_s)T(\theta_v)} \quad (5)$$

$$\rho_{\mathrm{ac}}=\frac{\rho'_{\mathrm{ac}}}{1+\rho'_{\mathrm{ac}}s} \quad (6)$$

ここで，λ の添え字をとった変数（E，T，T_g，s，ρ^a）は添え字が付いた変数に重み（E は $S(\lambda)$，それ以外は $S(\lambda)$ E_λ）を付けて平均した値であり，$\rho_{\mathrm{atm}}=\rho^a T_g$ である。

3. 2　6SV の実行

6SV のソースコードを gfortran 等の Fortran コンパイラでコンパイルすると sixsV 2.1 という名前の実行ファイルが作成される。以下では 6SV の実行ファイルが環境変数 PATH に含まれる場所（/bin）に置かれていると仮定して話を進める。6SV は入力パラメータを標準入力（通常はキーボード）から読み込むようになっており，以下のようにリダイレクトを使ってファイルから読み込ませることもできる。（記号 $ はコマンドプロンプトを表す。）

```
$ sixsV2.1 ＜input.dat
```

Code 3.1 に入力ファイルの例を示す。各パラメータの意味は Table 1 の通りである。設定によっては Table 1 にないパラメータも必要になる。6SV を実行すると，入力された設定値，放射伝達計算結果に続いて以下のような大気補正結果が標準出力（通常はディスプレイ）に出力される。

```
**************************************************
 atmospheric correction result
```

L	Value	Parameter
1	0	igeom
2	57.9 180.0 41.4 179.0 12 5	asol,phi0,avis, phiv,month,jday
3	3	idatm
4	2	iaer
5	0	v
6	0.05	taer55
7	0	xps
8	-1000	xpp
9	44	iwave
10	0	inhomo
11	0	idirec
12	0	igroun
13	0.1	ro
14	0	irapp
15	-0.3	rapp

Code 3.1　Sample input file of 6SV (input.dat)

```
 ----------------------------------
 input apparent reflectance : 0.300
 measured radiance [w/m2/sr/mic] : 105.286
 atmospherically corrected reflectance
 Lambertian case : 0.17038
 BRDF       case : 0.17038
 coefficients xa xb xc : 0.00393 0.23905 0.15608
 y=xa*(measured radiance)-xb;  acr=y/(1.+xc*y)
 coefficients xap xb xc : 1.380301 0.239055 0.156084
 y=xap*(measured reflectance)-xb;  acr=y/(1.+xc*y)
**************************************************
```

出力される大気補正係数（xap，xa，xb，xc）は，xap＝1/[T_g（θ_s, θ_v）T（θ_s）T（θ_v）]，xa＝xap×π/[$E\mu_s$]，xb＝xap×ρ_{atm}，xc＝s であり，式（4）〜式（6）より大気補正された地表面反射率（ρ_{ac}）は以下のように求められる。

$$y=\rho'_{\mathrm{ac}}=xap\times\rho_i^*-xb=xa\times L_{\mathrm{obs}}-xb \quad (7)$$

$$\rho_{\mathrm{ac}}=\frac{y}{1+xc\times y} \quad (8)$$

3. 3　標準大気モデルを使った大気補正

衛星画像の大気補正では，前述した大気補正係数をいろいろな条件で求める必要があるため，6SV に入力パラメータを渡して大気補正係数を受け取る関数があると便利である。そこで，Code 3.2 に示したように，太陽方向，視線方向，観測月日，エアロゾル光学的厚さを引数にとり，6SV を実行して得られる大気補正係数を出力する Python 関数（run_6s）を用意する。4〜19 行目で 6SV に入力したい内容を lines という変数に格納している。次に Popen モジュールを使って 6SV に標準入力から lines を入力し，標準出力，標準エラー出力をそれぞれ out，err という変数に

Table 1　Input parameters of 6SV

Parameter	Value
igeom	geometrical conditions (0: user-defined, 1-7: built-in satellite)
asol, phi0, avis, phiv, month, jday	solar zenith angle [deg] solar azimuth angle [deg] view zenith angle [deg] view azimuth angle [deg] month day of the month
idatm	atmospheric model (0: no absorption, 1-6: built-in model, 7, 8: user-defined)
iaer	aerosol type and profile (-1: user-defined profile, 0: no aerosols, 1-7: built-in model, 8-11: user-defined size distribution, 12: FILE)
v	aerosol concentration (-1: no aerosol, 0: enter optical depth, >0: visibility [km])
taer55	aerosol optical depth at 550nm
xps	target altitude (≤0: altitude [km], >0: pressure [hPa])
xpp	sensor altitude (-1000: satellite, 0: ground, -100<xpp<0: - xpp [km])
iwave	filter function (-2: enter min/max wavelength [μm], -1: enter single wavelen. [μm], 0: enter min/max wavelen. [μm], 1: enter min/max wavelen. [μm] and filter function by step of 2.5nm, 2-199: built-in satellite)
inhomo	ground reflectance type (0: homogeneous, 1: non uniform)
idirec	directional surface effect (0: no directional effect, 1: directional effect)
igroun	surface reflectance (-1: enter min/max wavelength [μm] and reflectance by step of 2.5nm, 0: enter constant, 1- 4: built-in model)
ro	reflectance
irapp	atmospheric correction mode (-1: no atmospheric correction, 0,1: atmospheric correction)
rapp	apparent reflectance/radiance (≥0: radiance [W/m^2/sr/μm], <0: reflectance = - rapp)

L	Code
1	`import re`
2	`from subprocess import Popen,PIPE`
3	`def run_6s(sza,saz,vza,vaz,mon, day,tau):`
4	`□lines = ''`
5	`□lines += '0¥n' # User defined`
6	`□lines += '{} {} {} {} {} {}¥n' .format(sza,saz,vza,vaz,mon,day)`
7	`□lines += '3¥n' # Midlat. Winter`
8	`□lines += '2¥n' # Maritime Model`
9	`□lines += '0¥n' # Constant tau`
10	`□lines += '{}¥n'.format(tau) #tau`
11	`□lines += '0¥n' # Target level`
12	`□lines += '-1000¥n' # Satellite`
13	`□lines += '44¥n' # MODIS Band 3`
14	`□lines += '0¥n' # Homogeneous`
15	`□lines += '0¥n' # No directional`
16	`□lines += '0¥n' # Constant rho`
17	`□lines += '0.1¥n' # rho`
18	`□lines += '0¥n' # Atm. correction`
19	`□lines += '-0.3¥n' # Reflectance`
20	`□p = Popen('/bin/sixsV2.1',stdin =PIPE,stdout=PIPE,stderr=PIPE)`
21	`□out,err = p.communicate (lines.encode())`
22	`□xap = xb = xc = None`
23	`□for line in out.decode() .splitlines():`
24	`□□m = re.search('coefficients¥s+ xap¥s+.*:¥s*(¥S+)¥s+(¥S+)¥s+(¥S+)' ,line)`
25	`□□if m:`
26	`□□□xap,xb,xc = [float(s) for s in m.groups()]`
27	`□return xap,xb,xc`

Code 3.2　Function to run 6SV

取り込んでいる。（入出力にファイルを介さない。）22 行目以降では，out の内容から大気補正係数（xap，xb，xc）を検索して見つかった値を return するようになっている。

簡単のため，ここでは 6SV の標準大気モデル（大気分子：Mid-latitude winter，エアロゾル：Maritime）を使用する。（カスタムモデルを使う方法は次回に説明する。）衛星センサの応答関数には，ひまわり 8 号のバンド 1（中心波長 0.47 μm）に相当する MODIS バンド 3 の応答関数を使用する。6SV に入力する ro や rapp の値は大気補正係数に影響しないので，ここでは適当な値（0.1 および−0.3）を入れておく。衛星センサで測定される地表面放射輝度から地表面反射率を導出するためには，エアロゾルに関する情報（光学的厚さ等）が必要である。ここでは空気が清浄な日（2015 年 12 月 5 日）について，光学的厚さが一様と仮定して千葉周辺地域の地表面反射率を導出する。光学的厚さの値には千葉大学環境リモートセンシング研究センター（CEReS，35.6246° N，140.1041° E）に設置された放射計（スカイラジ

L	Code
1	from datetime import datetime, timedelta
2	import ephem
3	import numpy as np
4	def calc_sunpos(tim,lon,lat):
5	□sun = ephem.Sun()
6	□obs = ephem.Observer()
7	□obs.date = tim
8	□obs.lat = np.radians(lat)
9	□obs.long = np.radians(lon)
10	□sun.compute(obs)
11	□return np.degrees(sun.az), 90.0-np.degrees(sun.alt), sun.earth_distance # 0<=azimuth<360d,north=0,cw
12	yr,mo,dy,hr,mi = 2015,12,5,11,30 # JST
13	tim = datetime(yr,mo,dy,hr,mi) +timedelta(hours=-9) # UTC
14	lon = 140.104128
15	lat = 35.624594
16	saz,sza,sdis = calc_sunpos(tim,lon,lat)

Code 3.3　Function to calculate solar position

L	Code
1	gain = 0.37735835
2	const = -7.5471671
3	coeff = 1.5588241e-03
4	R,r = 6371.0,42157.0
5	dxi = np.radians(140.7-lon)
6	gam = np.arccos(np.cos(np.radians (lat))*np.cos(dxi))
7	vza = np.degrees(np.arctan(r*np. sin(gam)/(r*np.cos(gam)-R)))
8	vaz = 180.0-np.degrees(np.arcsin (np.sin(dxi)/np.sin(gam)))
9	mu = np.cos(np.radians(sza))
10	lon,lat = np.meshgrid(np.arange(85.005,205.0,0.01),np.arange(59.995,-60.0,-0.01))
11	cnd = (lon>139.5) & (lon<141.0) & (lat>34.8) & (lat<36.0)
12	data = np.fromfile('201512050230.v is.01.fld.geoss',dtype='>u2') .reshape(12000,12000)[cnd] .reshape(120,150)
13	lobs = gain*data+const
14	rhoi = lobs*coeff*sdis*sdis/mu
15	xap,xb,xc = run_6s(sza,saz,vza, vaz,mo,dy,0.05)
16	y = xap*rhoi-xb
17	rho = y/(1.0+xc*y)

Code 3.4　Atmospheric correction example

オメーター）の測定値から推定された値（0.05 @550 nm）を用いる。

先に用意した関数（run_6 s）を使って 6SV を実行するためには，太陽方向および視線方向が必要である。Code 3.3 に示した関数（calc_sunpos）を使うと観測者の時刻 tim（UTC，datetime 形式），東経 ξ＝lon（度），北緯 η＝lat（度）を入力すれば太陽方位角 ϕ_s＝saz（度，北＝0，時計回り正），太陽天頂角 θ_s＝sza（度），太陽-地球間距離 D＝sdis（AU）が得られる。

また，経度 ξ_{sat} の赤道上空にある静止衛星を北半球の観測地点から見た時の天頂角 θ_v および方位角 ϕ_v は以下のように計算できる。

$$\gamma=\cos^{-1}\left[\cos\eta\cos(\xi_{sat}-\xi)\right] \tag{9}$$

$$\theta_v=\tan^{-1}\left[\frac{(R+h)\sin\gamma}{(R+h)\cos\gamma-R}\right] \tag{10}$$

$$\phi_v=180^\circ-\sin^{-1}\left[\frac{\sin(\xi_{sat}-\xi)}{\sin\gamma}\right] \tag{11}$$

ここで，R は地球半径（6371 km），h は衛星高度（35800 km），γ は地球中心から見た観測者と衛星の見込み角である。ひまわり 8 号の経度は 140.7° E なので，CEReS から見た場合，θ_v＝ 41.4°，ϕ_v＝ 179.0° となる。一方，CEReS から見た太陽の位置は南中時刻（2015 年 12 月 5 日 11：30 JST）において θ_s＝ 57.9°，ϕ_s＝ 180.0° である。

簡単のため，千葉周辺地域（東経 139.5°～141°，北緯 34.8°～36°）において衛星および太陽の方向が等しいとみなしてひまわり 8 号バンド 1 の地表面反射率を導出するコードを Code 3.4 に示す。ひまわり 8 号のデータには Gridded データ（201512050230.vis.01.fld.geoss）を用いる。前回はルックアップテーブルを使って Gridded データのラジオメトリック校正を行ったが，参考のため，今回はあえて別の方法で校正を行う。

Gridded データにはひまわり標準（HS）データと同じ 2 バイト符号なし整数（C）が格納されており，HS データから読み出した倍率（gain）およびオフセット（const）を使って

$$L_{obs}=\text{gain}\times C+\text{const} \tag{12}$$

のように放射輝度（単位は W/m^2/sr/μm）に変換できる。さ

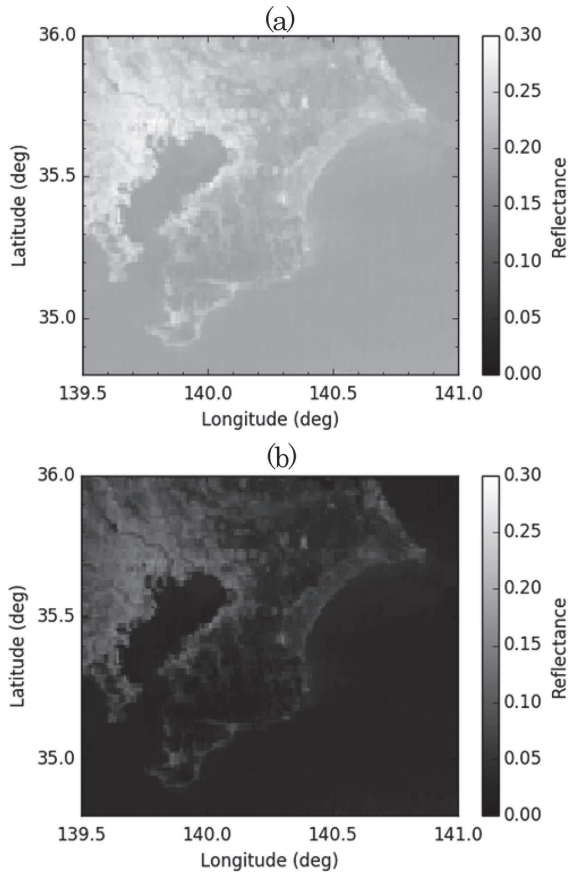

Fig. 2　Himawari-8 Band 1 image (a) before and (b) after atmospheric correction (Dec. 5, 2015 11 : 30 JST)

らに，HS データから読み出した係数（coeff）を使って

$$\rho_0 = \frac{\pi L_{\mathrm{obs}}}{E_0} = \mathrm{coeff} \times L_{\mathrm{obs}} \qquad (13)$$

のように反射率に変換できる。ここで，$E_0 = ED^2$（D は太陽-地球間距離，単位は AU）であり，式（4）より

$$\rho_i^* = \rho_0 \frac{D^2}{\mu_s} \qquad (14)$$

の関係があることが分かる。Fig. 2 に大気補正前（ρ_i*）および大気補正後（ρ_{ac}）の反射率を示す。大気補正によって反射率が全体的に小さくなっていることが見て取れる。

4.　おわりに

今回は Python を使って 6SV を動かし，ひまわり 8 号画像の大気補正を行う簡単な方法を示した。この方法を応用して，エアロゾルモデルに観測データを用いればより正確な大気補正が可能になる。また，空気が清浄な日に得られた地表面反射率を用いると，空気が比較的汚れた日のエアロゾル光学的厚さを導出することができる。これらの話題の詳細については次回に説明する。

引用文献

1）6SV ホームページ
http : //6 s.ltdri.org/（2016. 10. 24）
2）6SV User Manual Part 1
http : //6 s.ltdri.org/content/manual01.html（2016. 10. 24）
3）朝隈康司，美濃村満生，汝　飛剣，大堤新吾，久世宏明，竹内延夫, 千葉大学環境リモートセンシング研究センター研究報告集第 4 号　衛星データにおける大気補正と大気観測，千葉大学環境リモートセンシング研究センター /大気放射研究部門センサ研究分野，2000．http : //mitizane.ll.chiba-u.jp/metadb/up/assist1/C-05.pdf（2016. 10. 24）