連載講義

Python を利用した衛星画像の大気補正 第 3 回 カスタム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 3: Atmospheric Correction with Custom Atmospheric Models

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。最終回の今回は，分光放射計（スカイラジオメーター）で測定されたエアロゾル特性に基づくカスタムエアロゾルモデルを使って，ひまわり 8 号衛星画像の大気補正を行う。

2. エアロゾルモデルによる大気補正係数の違い

衛星で観測される見かけの放射輝度を L_{obs}，見かけの反射率（放射伝達コード 6SV に入力する反射率）を ρ_i* とすると，大気補正された反射率 ρ_{ac} は

$$y = \mathrm{xap} \times \rho_i^* - \mathrm{xb} = \mathrm{xa} \times L_{obs} - \mathrm{xb} \quad (1)$$

を使って

$$\rho_{ac} = \frac{y}{1 + xc \times y} \quad (2)$$

のように計算できる[1]。ここで，xap，xa，xb，xc は 6SV を実行して得られる大気補正係数であり，衛星・太陽の幾何学的条件，大気条件，エアロゾルモデル等，6SV の入力パラメータに依存する変数である。これら入力パラメータが空間的に一様と見なせない場合，衛星画像の画素毎に 6SV を実行すると時間がかかり過ぎるため，いくつかの入力パラメータについて式（2）の計算を行って作成したルックアップテーブル（LUT）を利用すると良い。Code 2.1 に波長 550 nm におけるエアロゾル光学的厚さ τ_{550} と ρ_i* を変数とする ρ_{ac} の LUT を作成するコードを示す。Python ではブロックの区切りにインデントを使うが，ここではスペース節約のために記号（□）を使い，2 個以上のインデントは□の中に個数を書くことにする。観測日時は 2015 年 12 月 5 日の 11：30 である。変数 τ_{550}（tau），ρ_i*（rhoi）とも，範囲は 0〜1，刻みは 0.1 に設定している。9 行目で前回作成した run_6s 関数を用いて大気補正係数を計算している。この関数の内部では大気モデルに Mid-latitude Winter，エアロゾルモデルに Maritime，波長帯に MODIS バンド 3（〜0.47 μm）を指定している。10〜16 行目で大気補正係数から ρ_{ac}（rho）を計算し，結果をファイルに書き出している。ここで得られた ρ_{ac} と ρ_i* および ρ_{ac} と τ_{550} の関係を図示すると Fig. 1（a）のようになる。この LUT を用いると，

L	Code
1	`import numpy as np`
2	`sza,saz = 57.9,180.0`
3	`vza,vaz = 41.4,179.0`
4	`mon,day = 12,5`
5	`tau = np.arange(0.0,1.01,0.1)`
6	`rhoi = np.arange(0.0,1.01,0.1)`
7	`rho = []`
8	`for t in tau:`
9	`□xap,xb,xc = run_6s(sza,saz,vza,vaz,mon,day,t)`
10	`□y = xap*rhoi-xb`
11	`□rho.append(y/(1.0+xc*y))`
12	`with open('lut.dat','w') as fp:`
13	`□for i in range(tau.size):`
14	`□2for j in range(rhoi.size):`
15	`□3fp.write(' {:10.3e}'.format (rho[i][j]))`
16	`□2fp.write('¥n')`

Code 2.1 Making Lookup Table

（2016. 12. 28 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

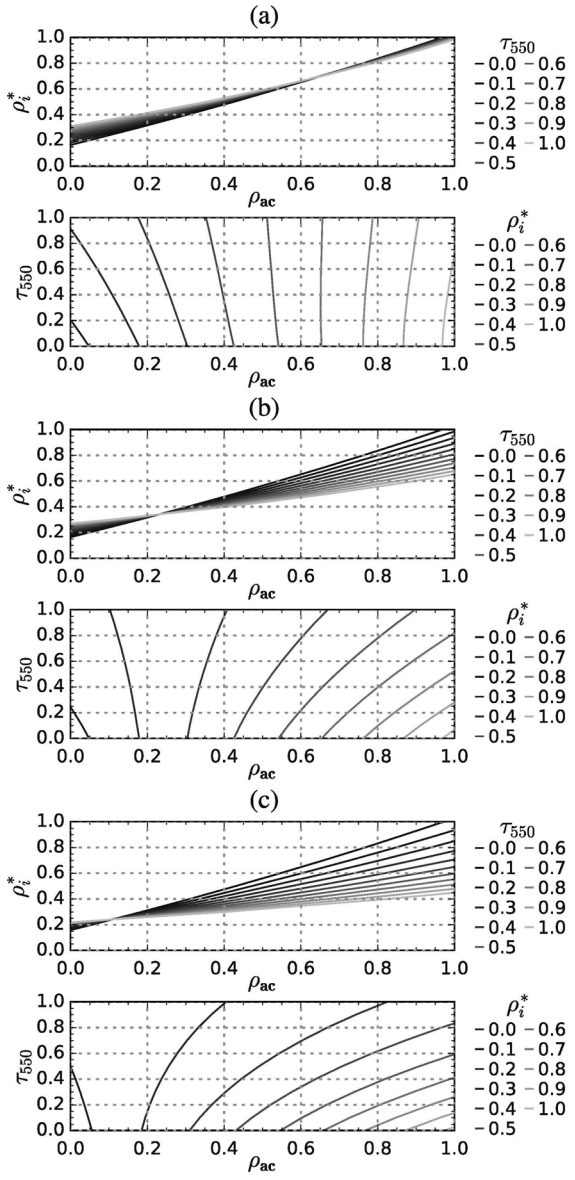

Fig. 1　Atmospherically corrected reflectance (ρ_{ac}) vs. apparent reflectance (ρ_i*) and aerosol optical depth (τ_{550}). (a) Maritime model (b) Continental model (c) Custom model

τ_{550} が分かれば ρ_{ac} と ρ_i* の関係から衛星画像の大気補正を行うことができ，逆に ρ_{ac} が分かっていれば τ_{550} を導出できる。（衛星画像から ρ_{ac} と τ_{550} を同時推定する方法も開発されている[2]。）

ここで，大気補正係数は仮定するエアロゾルモデルに強く依存することに注意する必要がある。Fig. 1（b）はエアロゾルに Continental モデルを仮定して得られた ρ_{ac} と ρ_i* および ρ_{ac} と τ_{550} の関係を示している。Fig. 1（c）は後述するカスタムエアロゾルモデルの結果である。Maritime モデルと比べると，ρ_{ac} が 0 に近い範囲では τ_{550} の影響が小さく，ρ_{ac} が 1 に近い範囲では τ_{550} の影響が大きいことが分かる。

したがって，大気補正の際はなるべく現実に即したエアロゾルモデル[3] を選ぶことが重要であり，可能ならば実測に基づくカスタムエアロゾルモデルを用いるのが良い。

3.　スカイラジオメーターのデータ読み出し

ここでは，地上設置の分光放射計（スカイラジオメーター）で測定されたエアロゾル特性を利用してカスタムエアロゾルモデルを作成する。スカイラジオメーターのデータは国立環境研究所の Web サイト[4] から標準プロダクト，千葉大学の Web サイト[5] から研究プロダクトが入手できるが，現状では粒径分布と複素屈折率の値が研究プロダクトにしか含まれていないため，研究プロダクトを利用する。

Code 3.1 にスカイラジオメーターの研究プロダクトを読み込む関数（read_skyrad）を示す。研究プロダクトのデータはヘッダー部とデータ部から成り，ヘッダー部の行数は先頭行に書かれている。ヘッダー部の最終行には “Year Month Day Hour（LT）DN（LT）AOT（0.340）… AOT（1.020）SSA（0.340）… SSA（1.020）dVdlnr（r＝0.012）… dVdlnr（r＝16.540）AngExp CloudFlag” のような項目が書かれており，12〜34 行目でこれらを読み込んで必要なパラメータの列番号，波長，粒径の値を取得している。これらが変わらないことが分かっているならば，定数値を与えてヘッダーを全て読み飛ばす（10 行目の n－1 を n にする）ことも可能である。データ部は 35 行目の 1 行だけで読み込める。（Python2 と Python3 の両方で動作するように，u″ を付けて Unicode 文字列に変換している。）36，37 行目では日時を扱いやすい datetime 形式に変換している。なお，現状では複素屈折率の値は研究プロダクトの previous version に含まれており，別途読み込む（Code 4.2 参照）。

4.　カスタムエアロゾルモデルを使った大気補正

これまでは大気補正に用いるエアロゾルモデルや衛星センサの応答関数に 6SV の組み込みモデルを利用したが，より正確に大気補正するためにはこれらをカスタマイズする必要がある。そこで Code 4.1 に示したように run_6 s 関数を修正する。元の run_6 s 関数と比べると，引数に粒径分布の半径（rad），粒径分布（vol），複素屈折率実部（refr），虚部（refi），応答関数の波長（wav），応答関数（rsr）が追加されている。複素屈折率は 6SV で決められた 20 波長に対応する値を与え，応答関数は 2.5 nm 間隔の波長（wav，単位は μm）に対応する値を与える必要がある。7〜17 行目で粒径分布と複素屈折率をセットし，22〜26 行目で応答関数をセットしている。応答関数セット後のコードは元の run_6 s 関数と変わらないため，28 行目以降は省略した。

Code 4.2 にカスタムエアロゾルモデル・応答関数を使って大気補正係数を計算するコードを示す。3〜14 行目ではスカイラジオメーターのデータ（研究プロダクトの previous version）を読み込み，1 次補間を使って 6SV で決められた 20 波長（w_ref）に対応する複素屈折率の実部

L	Code
1	from re import search
2	from datetime import datetime,timedelta
3	from io import StringIO
4	def read_skyrad(fnam):
5	□i_aot,i_ssa,i_vol = [],[],[]
6	□w_aot,w_ssa,rad = [],[],[]
7	□i_ang,i_cloud = None,None
8	□with open(fnam,'r') as fp:
9	②n = int(fp.readline())
10	②for i in range(n-1):
11	③fp.readline()
12	②pnam = fp.readline().split()
13	②for i,p in enumerate(pnam):
14	③m = search('AOT¥(([¥d¥.]+)¥)',p)
15	③if m:
16	④i_aot.append(i)
17	④w_aot.append(float(m.group(1)))
18	④continue
19	③m = search('SSA¥(([¥d¥.]+)¥)',p)
20	③if m:
21	④i_ssa.append(i)
22	④w_ssa.append(float(m.group(1)))
23	④continue
24	③m = search('¥(r=([¥d¥.]+)¥)',p)
25	③if m:
26	④i_vol.append(i)
27	④rad.append(float(m.group(1)))
28	④continue
29	③if search('AngExp',p):
30	④i_ang = i
31	④continue
32	③if search('CloudFlag',p):
33	④i_cloud = i
34	④continue
35	②data = np.loadtxt(StringIO (u''+fp.read()),unpack=True)
36	□yr,mo,dy = np.int32(data[[0,1,2]]+0.5)
37	□dtim = np.array([datetime(y,m,d) +timedelta(hours=h) for y,m,d,h in zip(yr,mo,dy,data[3])])
38	□return dtim,w_aot,data[i_aot], w_ssa,data[i_ssa],rad,data[i_vol], data[i_ang],data[i_cloud]

Code 3.1　Reading Skyradiometer research product

L	Code
1	from subprocess import Popen,PIPE
2	def run_6s(sza,saz,vza,vaz,mon,day ,rad,vol,refr,refi,tau,wav,rsr):
3	□lines = ''
4	□lines += '0¥n' # User defined
5	□lines += '{} {} {} {} {} {}¥n' .format(sza,saz,vza,vaz,mon,day)
6	□lines += '3¥n' # Midlat. Winter
7	□lines += '11¥n' # SunPhoto Model
8	□lines += '{}¥n'.format(vol.size)
9	□for r,v in zip(rad,vol):
10	②lines += '{} {}¥n'.format(r,v) # rad in um, vol in cm3/cm2/um
11	□for r in refr:
12	②lines += ' {}'.format(r)
13	□lines += '¥n'
14	□for r in refi:
15	②lines += ' {}'.format(r)
16	□lines += '¥n'
17	□lines += '0¥n' # Not saved
18	□lines += '0¥n' # Constant tau
19	□lines += '{}¥n'.format(tau) #tau
20	□lines += '0¥n' # Target level
21	□lines += '-1000¥n' # Satellite
22	□lines += '1¥n' # User's filter
23	□lines += '{} {}¥n'.format(wav[0] ,wav[-1]) # Min and Max wav in um
24	□for r in rsr:
25	②lines += ' {}'.format(r)
26	□lines += '¥n'
27	□lines += '0¥n' # Homogeneous
	The rest is omitted.

Code 4.1　Function to run 6SV with custom aerosol model and filter

（mr），虚部（mi）を求めている。観測時刻は mr＞1.2，mi＞0 を満たす波長数が 7 個ある条件で 2015 年 12 月 5 日 11：30 JST に最も近いものを選んでいる。15〜18 行目ではスカイラジオメーターのデータ（研究プロダクトの current version）を読み込み，6SV に入力するエアロゾル光学的厚さおよび粒径分布を求めている。19〜28 行目ではひまわり 8 号の Web サイト[6] から取得したデータ（エクセルシートにある Band 1 のデータを band01.txt というテキストファイルに保存した）を読み込み，波長間隔 2.5 nm の応答関数を求めている。

エアロゾル光学的厚さを変えて Code 2.1 と同様の計算

L	Code
1	from scipy.interpolate import splrep,splev
2	t0 = datetime(2015,12,5,11,30)
3	v = np.loadtxt('pom02p_lv2_chiba2015a.txt',skiprows=71,unpack=True)
4	mo,dy = np.int32(v[[1,2]]+0.5)
5	t1 = np.array([datetime(2015,m,d)+timedelta(hours=h) for m,d,h in zip(mo,dy,v[3])])
6	mr,mi = v[6:17],v[17:28]
7	j = np.where(((mr>1.2)&(mi>0)).sum(axis=0) == 7)[0]
8	i = j[np.argmin(np.abs(t1[j]-t0))]
9	t0,mr,mi = t1[i],mr[:,i],mi[:,i]
10	w = np.array([.315,.34,.38,.4,.5,.675,.87,.94,1.02,1.627,2.2])
11	w_ref = np.array([.35,.4,.412,.443,.47,.488,.515,.55,.59,.633,.67,.694,.76,.86,1.24,1.536,1.65,1.95,2.25,3.75])
12	cndr,cndi = (mr > 1.2),(mi > 0)
13	mr = splev(w_ref,splrep(w[cndr],mr[cndr],k=1)).clip(1.2,2)
14	mi = splev(w_ref,splrep(w[cndi],mi[cndi],k=1)).clip(0,1)
15	t2,wa,aot,ws,ssa,rad,vol,ang,cloud = read_skyrad('skyradio_chiba2015_s02c00_v05.00.00.txt')
16	i = np.argmin(np.abs(t2-t0))
17	tau = splev(0.55,splrep(wa,aot[:,i]))
18	vol = vol[:,i]
19	wv,v = np.loadtxt('band01.txt',usecols=(0,2),unpack=True)
20	wr,rsr = [],[]
21	ws,wd = 0.0025,0.00125
22	wmin = np.floor(wv.min()/ws)*ws
23	wmax,w = wv.max(),wmin
24	while w < wmax:
25	□cnd = (wv >= w-wd) & (wv < w+wd)
26	□wr.append(w)
27	□rsr.append(v[cnd].mean())
28	□w += ws
29	xap,xb,xc = run_6s(sza,saz,vza,vaz,mon,day,rad,vol,mr,mi,tau,wr,rsr)

Code 4.2　Preparation for input data of run_6s function

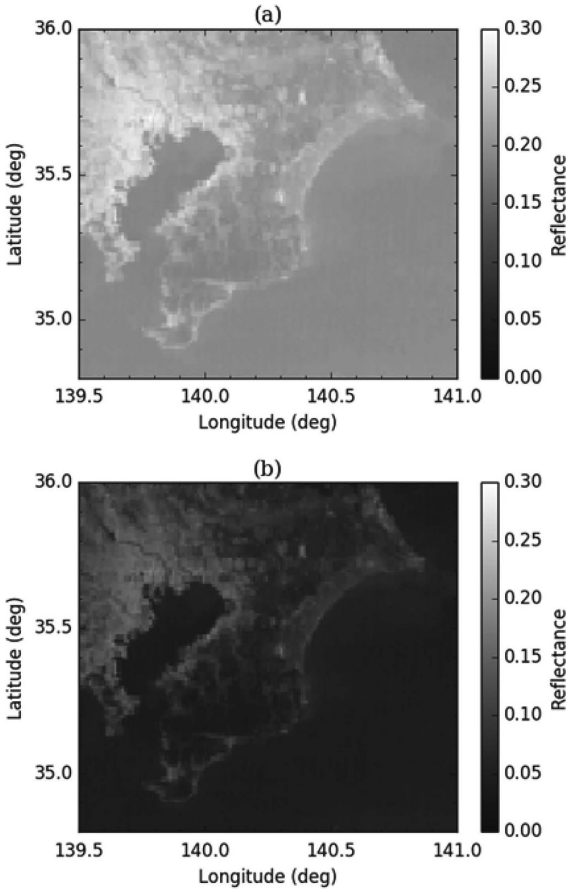

Fig. 2　Himawari-8 Band 1 image (a) before and (b) after atmospheric correction with custom aerosol model and response function (Dec. 5, 2015 11:30 JST)

を行うことによって得られた ρ_{ac} と ρ_i* および ρ_{ac} と τ_{550} の関係を図示すると Fig. 1(c) のようになった。また，画像全体のエアロゾル光学的厚さがスカイラジオメーターの測定値に等しいと仮定して大気補正すると Fig. 2 のような結果が得られた。

5.　エアロゾル光学的厚さの導出

前節では大気が清浄でエアロゾル光学的厚さ（τ_{550}）が小さい日（2015 年 12 月 5 日 11：30JST，τ_{550}＝0.027 @ 千葉大学）に取得された衛星画像の大気補正を行い，地表面反射率の真値（ρ_{ac}）を求めた。この ρ_{ac} を利用して，同時期で大気が混濁した日（2015 年 12 月 9 日 11：30JST，τ_{550}＝0.085 @ 千葉大学）のエアロゾル光学的厚さを導出する。Fig. 3 (a) に大気混濁時のひまわり 8 号衛星画像（バンド 1）を示す。Fig. 2 (a) と比較すると全体的に見かけの反射率（ρ_i*）が大きく，右下には雲が見られる。

Code 5.1 にエアロゾル光学的厚さを導出するためのコードを示す。2〜14 行目では大気が混濁した日の条件で ρ_{ac} (rho) の LUT を作成している。この日のスカイラジオメーターのデータは Code 4.2 と同様にしてあらかじめ読み込

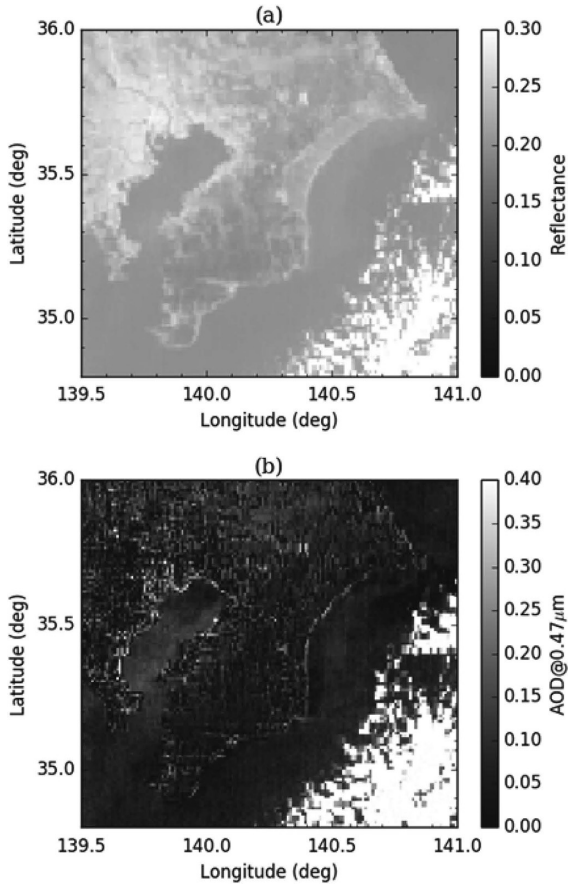

Fig. 3　(a) Himawari-8 Band 1 image without atmospheric correction (b) Aerosol optical depth retrieved with custom aerosol model and response function (Dec. 9, 2015 11:30 JST)

んでおく。15〜20 行目ではひまわり 8 号の Gridded データを読み込み，ρ_i*（ri）を計算している。21 行目では空気が清浄な日に得られた ρ_{ac}（ro）を読み込んでいる。22 行目では griddata 関数を使って LUT から ro，ri に対応する τ_{550}（tau_out）を計算している。（1 組の ρ_{ac}，ρ_i* に対応する τ_{550} が複数ある場合はもう少し工夫が必要。）得られたエアロゾル光学的厚さ（tau_out）は Fig. 3（b）のように右下の雲領域を除いてほぼ一様に分布している。ここではスカイラジオメーターの測定に基づくエアロゾルモデルを使用しており，光学的厚さがあまり大きくない（＜1）エアロゾルを対象としているため，厚い雲がある領域では光学的厚さを正確に導出できないことに注意が必要である。

6.　おわりに

以上，3 回に渡って Python を使って衛星画像の基礎的な大気補正に必要なデータ処理を行うための解説を行った。Python コードをほぼ省略せずに示したためにコードの記述が多くなったが，Python を使えば C 言語や Fortran 等のプログラミング言語に比べていかにデータ処理が容易であるかを伝えることができたならば幸いである。

L	Code
1	from matplotlib.mlab import griddata
2	sza,saz,sdis = 58.4,179.6,0.9850
3	vza,vaz = 41.4,179.0
4	mon,day = 12,9
5	mu = np.cos(np.radians(sza))
6	r = np.arange(0.0,1.001,0.01)
7	t = np.arange(0.0,0.501,0.01)
8	rho = []
9	for tau in t:
10	□xap,xb,xc = run_6s(sza,saz,vza, vaz,mon,day,rad,vol,refr,refi,tau, w_rsr,rsr)
11	□y = xap*r-xb
12	□rho.append(y/(1.0+xc*y))
13	rho = np.array(rho)
14	rhoi,tau = np.meshgrid(r,t)
15	lon,lat = np.meshgrid(np.arange(85.005,205.0,0.01),np.arange(59.995,-60.0,-0.01))
16	cnd =(lon>139.5) & (lon<141.0) & (lat>34.8) & (lat<36.0)
17	data = np.fromfile('201512090230.vis.01.fld.geoss',dtype='>u2') .reshape(12000,12000)[cnd] .reshape(120,150)
18	gain,const,coeff = 0.37735835, -7.5471671,1.5588241e-03
19	lobs = gain*data+const
20	ri = lobs*coeff*sdis*sdis/mu
21	ro = np.load('rho.npy')
22	tau_out = griddata(rho.flatten(), rhoi.flatten(),tau.flatten(), ro,ri,interp='linear')

Code 5.1　Retrieval of aerosol optical depth

謝　辞：ひまわり 8 号の Gridded データおよびスカイラジオメーターの研究プロダクトは千葉大学環境リモートセンシング研究センター（CEReS）から取得したものを使わせていただきました。

引用文献

1）6SV User Manual Part 1

http : //6 s.ltdri.org/content/manual01.html
（2016. 12. 26）
2）Y. Iikura, et al.: Surface reflectance estimation from satellite imagery with inhomogeneous atmospheric conditions, IGARSS 2015, pp.2131−2134 2015.
3）N. Manago, et al.: Seasonal variation of tropospheric aerosol properties by direct and scattered solar radiation spectroscopy, JQSRT 112 (2), pp. 285−291 2011.
4）SKYNET International Data Center
http : //www-lidar.nies.go.jp/skynet/index.php (2016. 12. 26)
5）SKYNET Research Product Release Service
http : //atmos2.cr.chiba-u.jp/skynet/data.html
(2016. 12. 26)
6）放射計（AHI）の応答関数
http : //www. data. jma. go. jp/mscweb/ja/himawari89/space_segment/spsg_ahi.html（2016. 12. 26）