UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

DEPARTMENT OF SPACE AND APPLICATIONS

DATA ACQUISITION AND SATELLITE SENSORS

LECTURERS: *Dr. Tong Si Son, Dr. Pham Duc Binh*

FINAL EXAM REPORT

Author:

Duong Thu Phuong

Student ID:

22BI13362

November 14, 2024

Contents

1 Data Acquisition

1.1 Collecting data

Data can be acquired from many satellites from various websites:

NASA Landsat Science

Corpernicus browser

There are Sentinel-1, Sentinel-2, Landsat 8 , etc...

And we focus mainly on Landsat 8 data

1.2 Pre-processing

After getting data of our location, we can start to process the data.

The data we get need to be corrected: radiometric correction, geometric correction, atmospheric correction, etc...

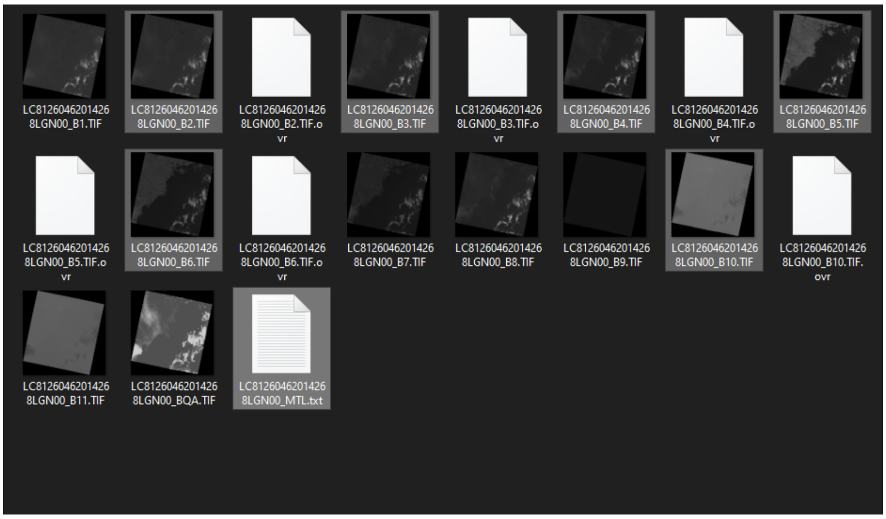


Figure 1: Data before corrected

1.3 Data processing

1.3.1 Collocation

After choosing all the needed bands, we collocate them, the process does two things at the same time: resamples the data into the same spatial resolution and puts them into one product.
Here I use nearest neighbour resampling method.

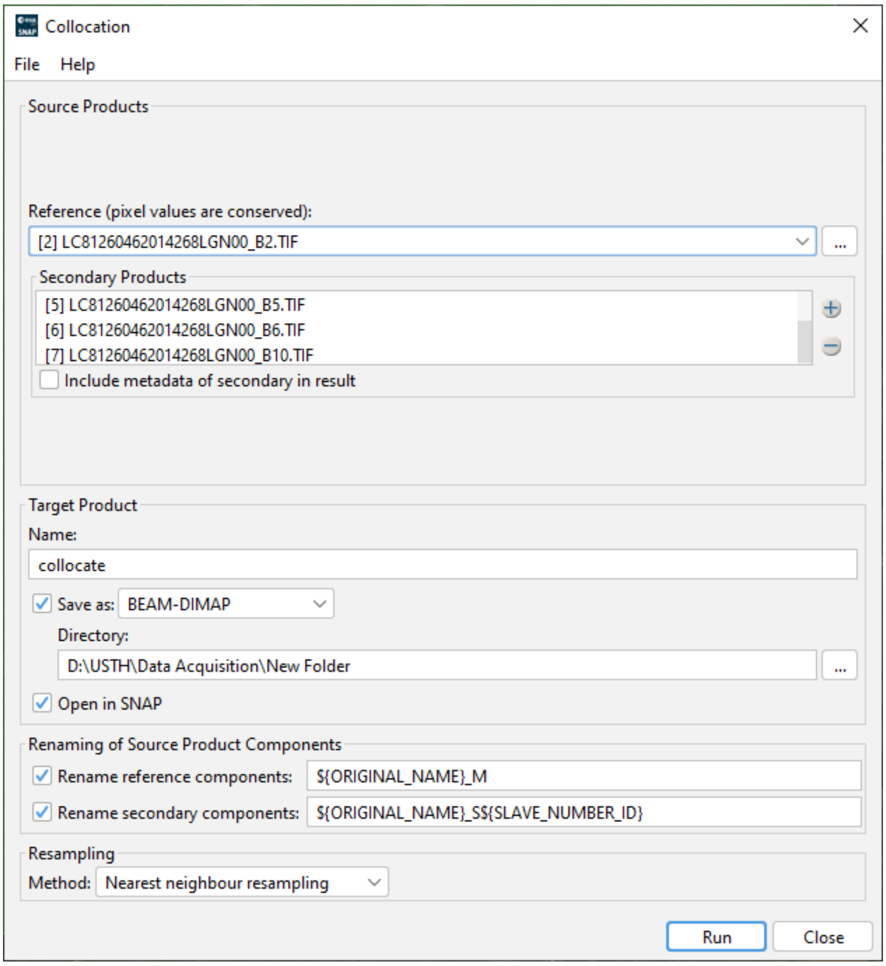


Figure 2: Collocation method

We choose B2 and it becomes the Master band, all secondary products become Slave bands.

1.3.2 RGB image

Opening the product, we retrieve RGB image by "Open RGB windows".

After collocating, the bands 2, 3, 4, 5, 6, 10 become M, S0, S1, S2, S3, S4

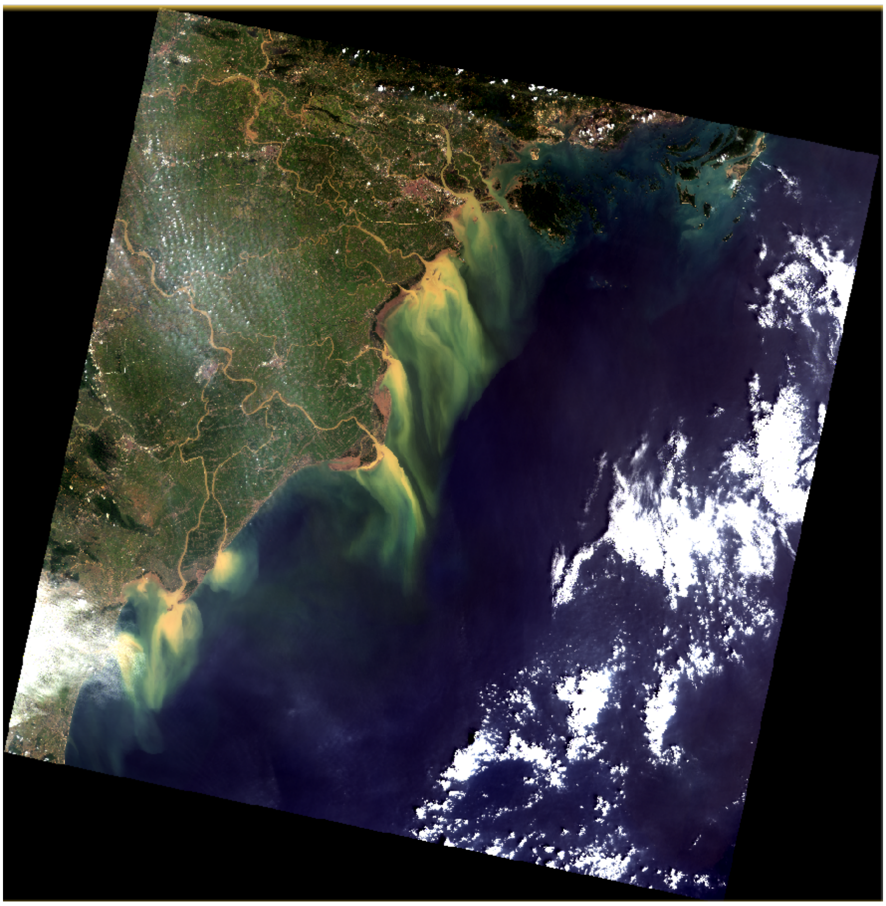


Figure 3: RGB image

1.3.3 On top of atmosphere

The bands we get are in Digital Number, we have to convert them back to energy adn then temperature through metadata and equations:

$$L_\lambda = M_L Q_{cal} + A_L \tag{1}$$

where:

1. L_λ is TOA radiance (W/ ($m^2 * srad * \mu m$))
2. M_L is Radiance multiply band
3. A_L is Reflectance add band

The band we use should be thermal band, which is either band 10 or 11.

$$T_B = \frac{K_2}{\ln(\frac{K_1}{L_\lambda} + 1)} \tag{2}$$

where

1. T_B is Top Of Atmosphere brightness temperature (K)
2. $K_1,\ K_2$ are band-specific thermal conversion constants

1.3.4 Choosing BL

We won't need to process the whole image; instead, we only do my part BL13.

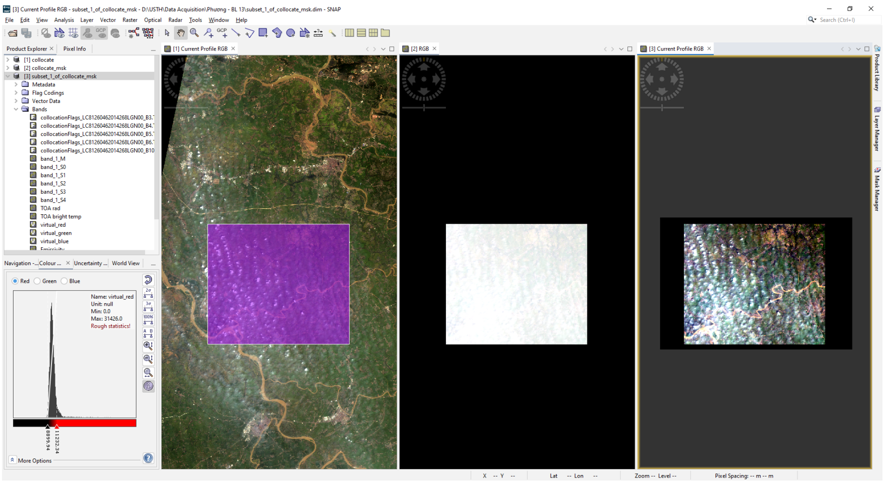


Figure 4: Subset

We mask out all other pixels except for my region, re-open RGB image, and subset it.

1.3.5 Emissivity map

Calculating the Land Surface Temperature needs the emissivity of cover land.

For the scope of this report, we use 4 vector data.

They are cloud with emissivity value 0.95, water 0.995, concrete 0.938, and forest 0.982.

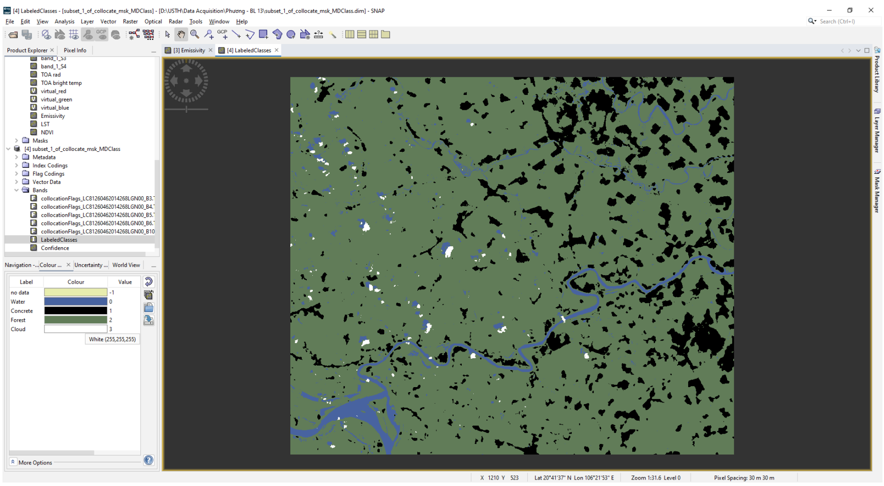


Figure 5: Emissivity map

1.3.6 LST

After getting the emissivity, we use the equation to calculate the Land Surface Temperature:

$$T = \frac{T_B}{1 + (\lambda * \frac{T_B}{C2}) * \ln(\varepsilon)} \tag{3}$$

where

1. T_B: Top of atmosphere brightness temperature
2. λ: wavelength of emitted radiance
3. $C2 = \frac{h*c}{s} = (1.4388 * 10^{-2}$ m K
4. h = Planck's constant= $6.626 * 10^{-34}$ J s
5. s = Boltzmann's constant = $1.38 * 10^{-23}$ J/K
6. c = velocity of light = $2.988 * 10^{8}$ m/s
7. ε: emissivity values of various land cover types

The wavelength is from 10.60 μm to 11.19 μm for band 10 in Landsat 8 satellites.

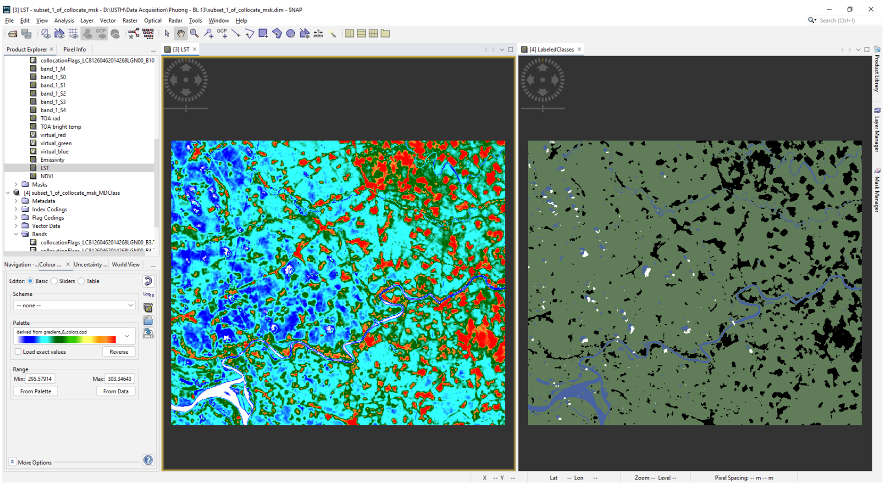


Figure 6: LST and Emissivity

1.3.7 NDVI

To see how much vegetation cover, we calculate NDVI:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{4}$$

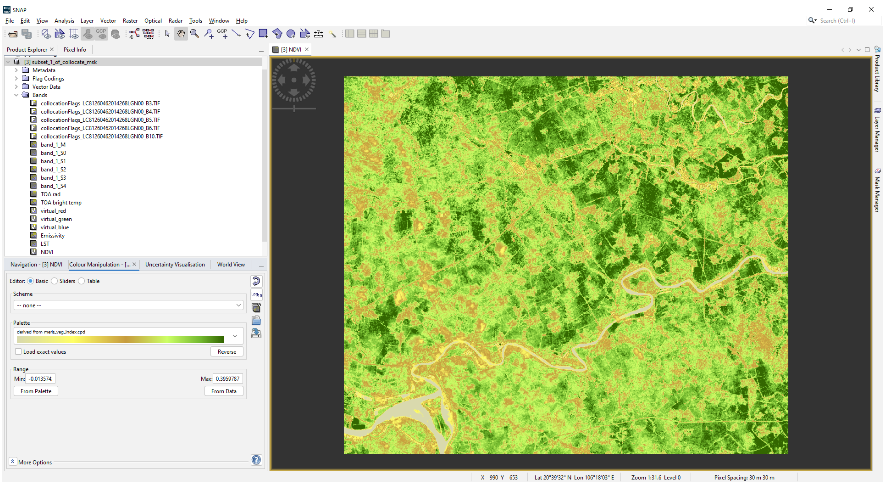


Figure 7: NDVI

More green, more plants.

1.3.8 NDVI vs LST

Plotting NDVI agaisnt LST, we get the scatter plot:

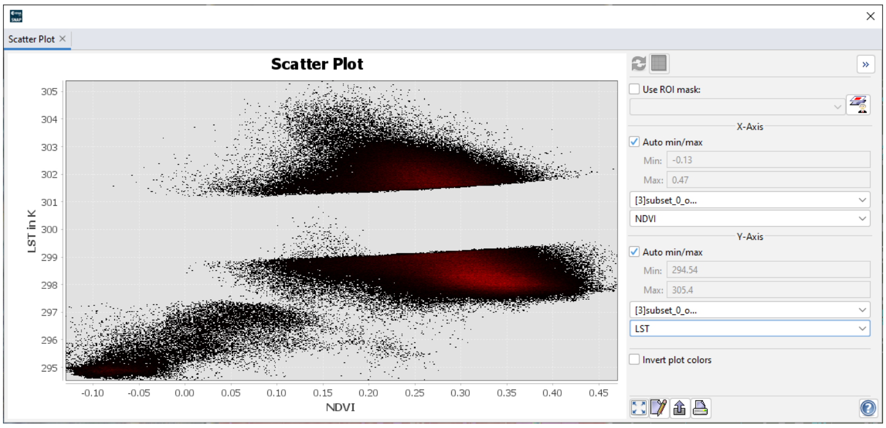


Figure 8: Scatter plot

1. Positive Aspects:
   1. Clear triangular/trapezoidal shape which is expected for NDVI-LST relationship
   2. NDVI: -0.12 to 0.47 (appropriate for mixed land cover)
   3. LST: 294.51K to 308.52K (≈ 21.4°C to 35.4°C)
2. Clusters:
   1. Three distinct clusters are visible:
   2. LST ≈ 295-297K (Water bodies)
   3. LST ≈ 298-300K (Vegetation areas)
   4. LST ≈ 301-306K (Built-up/bare areas)
3. Areas for Attention:
   1. The gap in the middle (around 299-301K)

1.3.9 Soil Moisture Index

From scatter plot, we do linear regression to get the warm edge and cold edge.

For our data, approximate values are:

$$\begin{aligned} \mathrm{LST}_{\mathrm{max}} &\approx 303.5 - 3.5 \times \mathrm{NDVI} \\ \mathrm{LST}_{\mathrm{min}} &\approx 296.0 + 1.5 \times \mathrm{NDVI} \end{aligned} \tag{5}$$

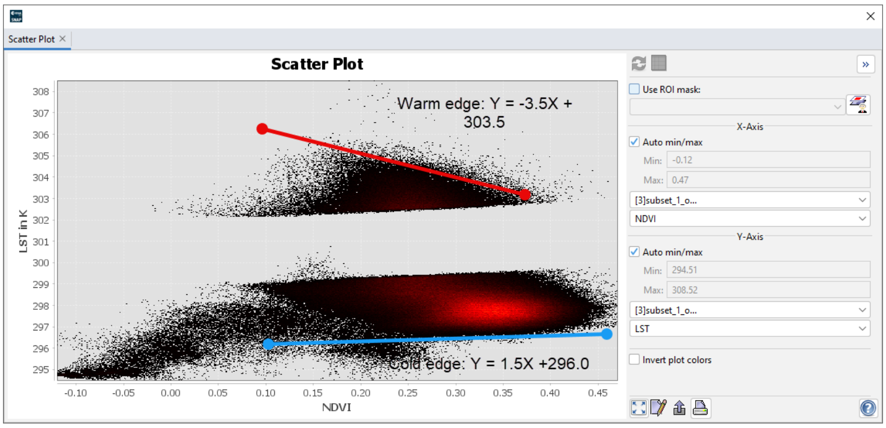


Figure 9: Warm edge and cold edge of scatter plot

Calculate Soil Moisture Index with equation:

$$SMI = \frac{LST_{max} - LST}{LST_{max} - LST_{min}} \tag{6}$$

And then compare with the Land Surface Temperature image.

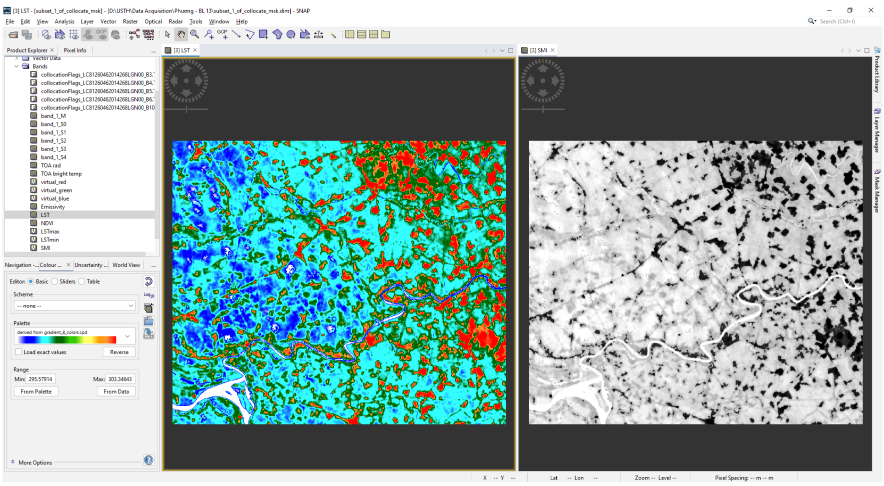


Figure 10: LST and SMI

Some observations from the comparision:

1. Spatial patterns
    - The river channel is clearly visible in both images
    - Shows strong inverse relationship between LST and SMI
    - Areas near the river show: Lower temperatures (blue in LST) ; Higher soil moisture (brighter in SMI)
2. Interpretation
    - River and riparian zones show expected patterns:
        - Cool temperatures
        - High moisture content
    - Upland areas show:
        - Higher temperatures (red/pink)
        - Lower soil moisture (darker tones)

1.4 Conclusion

The practice successfully demonstrated the strong interconnection between land surface temperature (LST), vegetation coverage (NDVI), and soil moisture conditions (SMI) in the study area.

The analysis conclusively demonstrates that remote sensing is a reliable and effective tool for environmental observation. The results clearly show the intricate relationships between three key environmental parameters: ground surface temperatures, plant coverage patterns, and soil water content, particularly in areas where urban development meets natural landscapes. This interconnected relationship is especially evident in the transition zones between city and surrounding areas, reinforcing the value of satellite-based monitoring for understanding complex environmental dynamics.

2 Satellite Sensors

2.1 Working with water level data

After getting data from HydroWeb and G-Realm of the Tonle Sap Lake in Cambodia, we begin to process the data.

Figure 11: HydroWeb data

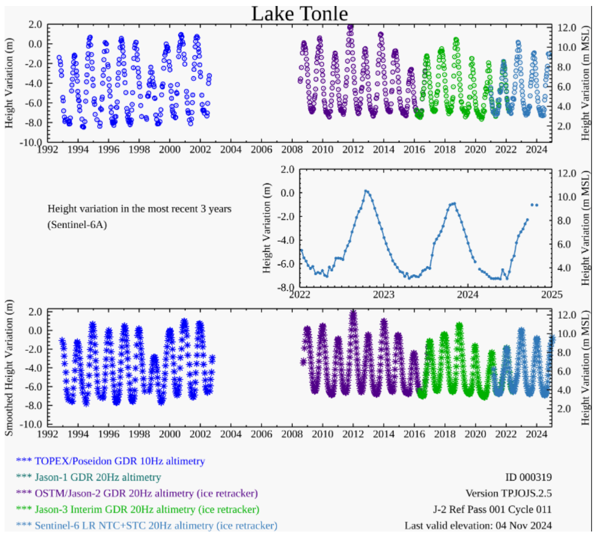


Figure 12: G-Realm data

Using Python, we retrieve the data from txt file to time series from 2012 to the end of 2017.

```
tonle_realm=pd.read_csv('height_tonle.txt',sep='\s+',header=None,skiprows=15,
    na_values=['9999999', '99.99','999.99','9999.99'])
tonle_realm=tonle_realm[[0,4]]
tonle_realm.columns=['Date','Height']
tonle_realm.isnull().sum()
tonle_realm.dropna(inplace=True)
tonle_realm['Date'] = pd.to_datetime(tonle_realm['Date'], format='%Y-%m-%d')
tonle_realm.head()
tonle_realm_area = tonle_realm[(tonle_realm['Date'] >= '2012-01-01') & (
    tonle_realm['Date'] <= '2019-12-31')]
tonle_realm_area.set_index('Date', inplace=True)
tonle_realm_area.head()

plt.figure(figsize=(18,6))
plt.plot(tonle_realm_area.index, tonle_realm_area['Height'], color='skyblue')
plt.xlabel('Date')
plt.ylabel('Height (m)')
plt.title('G-Realm data for Tonle Sap Lake Water Level')
plt.grid()
```

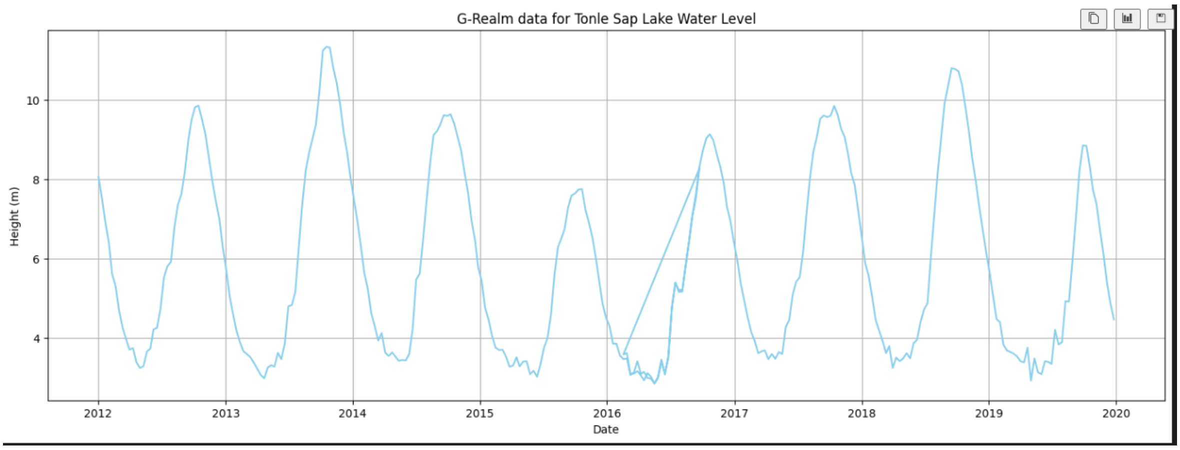


Figure 13: G-Realm time series

Adding the data from Hydroweb for comparison:

```python
tonle_hydroweb=pd.read_csv('plot.csv',sep=',',header=None,skiprows=1)

tonle_hydroweb.isnull().sum()
tonle_hydroweb=tonle_hydroweb[[0,1]]
tonle_hydroweb['Date'] = pd.to_datetime(tonle_hydroweb['Date'], format='%Y-%m-%d')
tonle_hydroweb_area = tonle_hydroweb[(tonle_hydroweb['Date'] >= '2012-01-01') & (tonle_hydroweb['Date'] <= '2019-12-31')]
tonle_hydroweb_area.set_index('Date', inplace=True)
tonle_hydroweb_area.head()

plt.figure(figsize=(18,6))
plt.plot(tonle_hydroweb_area.index, tonle_hydroweb_area['Height'],color='red', label='hydroweb')
plt.xlabel('Date')
plt.ylabel('Height (m)')
plt.title('G_Realm and Hydroweb data for Tonle Sap Lake Water Level')
plt.legend()
plt.grid()
```

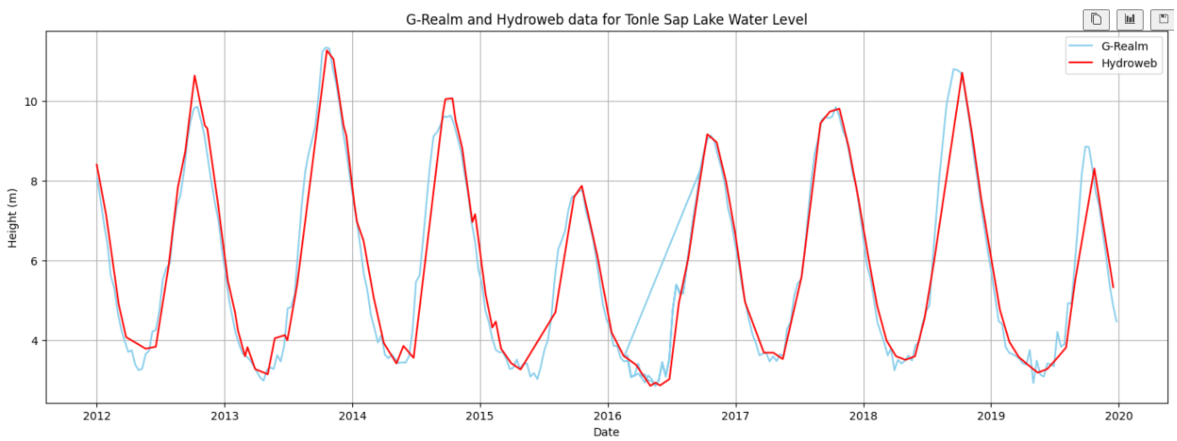


Figure 14: G-Realm and Hydroweb data

We resample the G-Realm data to get the monthly data to match with HydroWeb.

```
tonle_realm_area_monthly = tonle_realm_area.resample('M').mean()
tonle_hydroweb_area_monthly = tonle_hydroweb_area.resample('M').mean()
tonle_realm_area_monthly.head()
tonle_hydroweb_area_monthly.head()

plt.figure(figsize=(18,6))
plt.plot(tonle_realm_area_monthly.index, tonle_realm_area_monthly['Height'], color='skyblue', label='G-REALM')
plt.plot(tonle_hydroweb_area_monthly.index, tonle_hydroweb_area_monthly['Height'],color='red', label='hydroweb')
plt.xlabel('Date')
plt.ylabel('Height (m)')
plt.legend()
plt.grid()
```

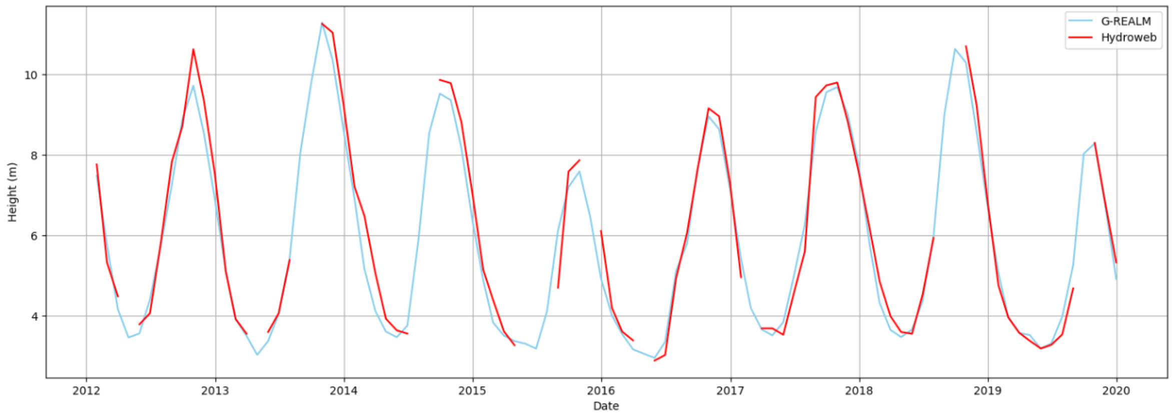


Figure 15: Match data

```
    tonle_grealm_area_monthly.corrwith(tonle_hydroweb_area_monthly)
    tonle_grealm_area.mean()
    tonle_hydroweb_area.mean()
```

We get the mean of G-realm water level = 5.74454 (m) and of HydroWeb = 5.888077 (m).

The correlation between 2 time series is R = 0.985136, very positive.

Combining the mean of 2 time series, we get:

```
tonle_df = pd.concat([tonle_grealm_area_monthly['Height'], tonle_hydroweb_area_monthly['Height']], axis=1)

tonle_mean = tonle_df.mean(axis=1)
tonle_std = tonle_df.std(axis=1)

plt.figure(figsize=(18, 6))
plt.plot(tonle_mean, label='Mean')
plt.fill_between(tonle_mean.index, tonle_mean - tonle_std, tonle_mean + tonle_std, color='gray', alpha=0.5)
plt.xlabel('Date')
plt.ylabel('Height (m)')
plt.legend()
plt.grid()
```

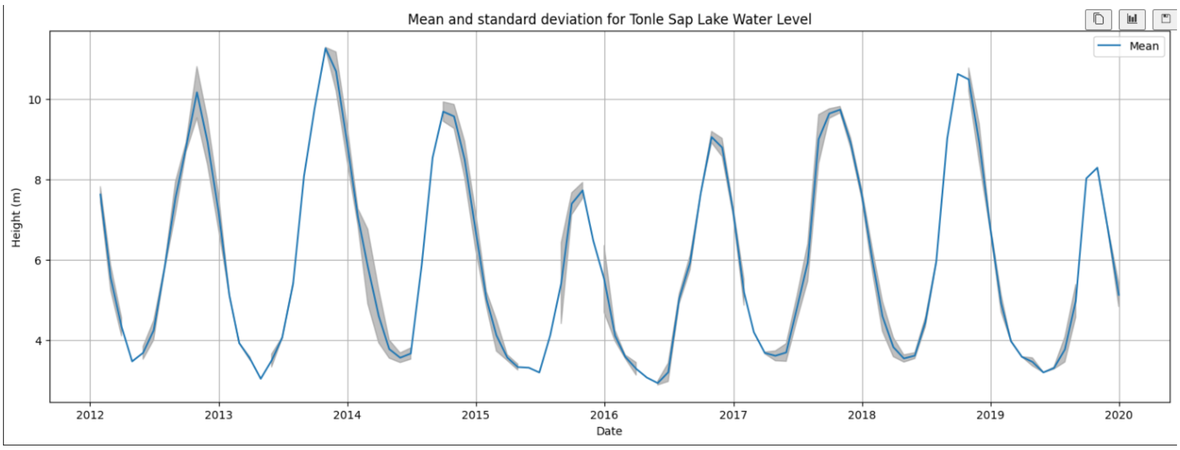
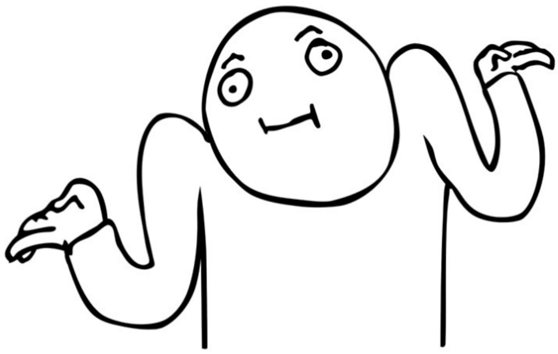


Figure 16: Mean and standard deviation

2.2 Working with MODIS images to get the time series of water extent of the lake



Appendix

List of Figures