THE UNIVERSITY OF ARIZONA

BE/BAT 485/585

Remote Sensing Data and Methods

Lab - 10

Instructor: Kamel Didan[1,2]

Helpers: Dr. Armando Barreto[1,2]

Mr. Truman Combs[1,2]

[1]BE Dept., University of Arizona,

[2]VIP Lab.

vip.arizona.edu

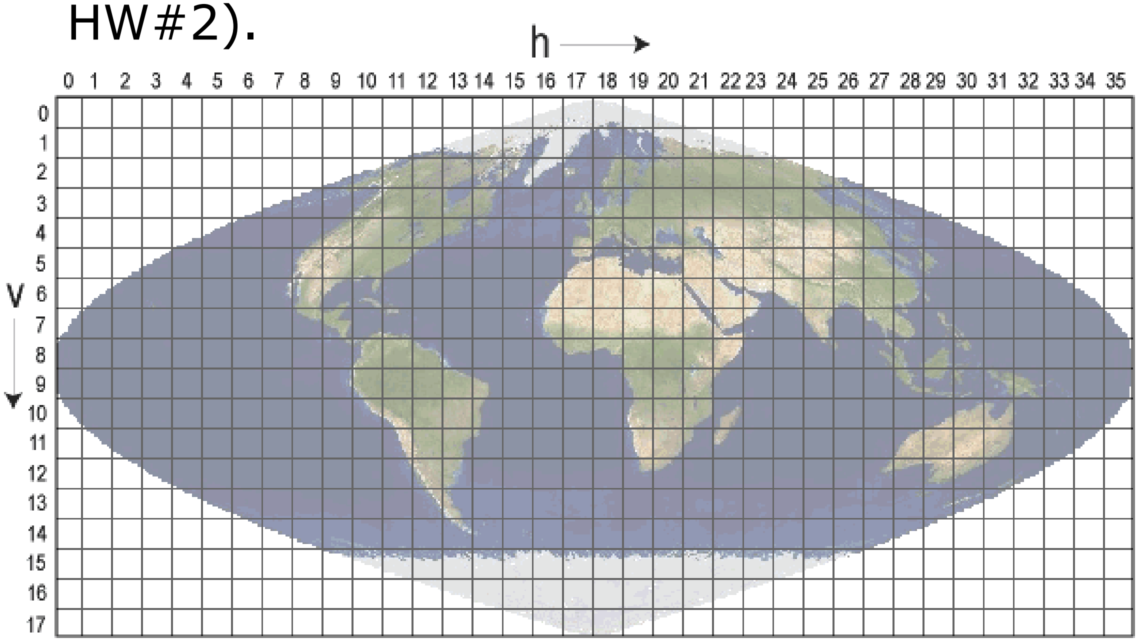
vegetation index & phenology Lab.

...Understanding a piece of the Earth system

MODIS

MODERATE RESOLUTION IMAGING SPECTRORADIOMETER

- MODIS is a key instrument (sensor) aboard [Terra](#) (original name EOS AM-1) and [Aqua](#) (EOS PM-1) satellites. AM and PM denote morning and afternoon overpass. Original plans for 5+5 platforms
- Terra's orbit around the Earth is timed so that it passes from North to South across the equator in the morning, while Aqua passes South to North over the equator in the afternoon.
- Terra & Aqua MODIS (identical sensors) view the entire Earth's surface every 1 to 2 days, and acquire data in 36 spectral bands, or groups of wavelengths (search MODIS Technical Specifications for more info – HW#2).



Most MODIS datasets are generated and made public as tiles (10x10 degrees squares ~ 1200 km x 1200 km), in the **Sinusoidal** projection and HDF4/HDF5 file format.

MODIS vegetation index (VI) data

The MODIS VI products (MOD13), are a Level 3 products, and like all other MODIS land products, provide consistent, spatial and temporal comparisons of global vegetation conditions which can be used to monitor the Earth's terrestrial photosynthetic vegetation activity in support of phenologic, change detection, and biophysical interpretations.

MODIS File naming convention:

MOD13A2.A2020065.h08v05.006.2020082012602.hdf

MOD:Terra
MYD:Aqua

YEAR

DOY

TILE

VERSION

PRODUCTION
DATE & TIME

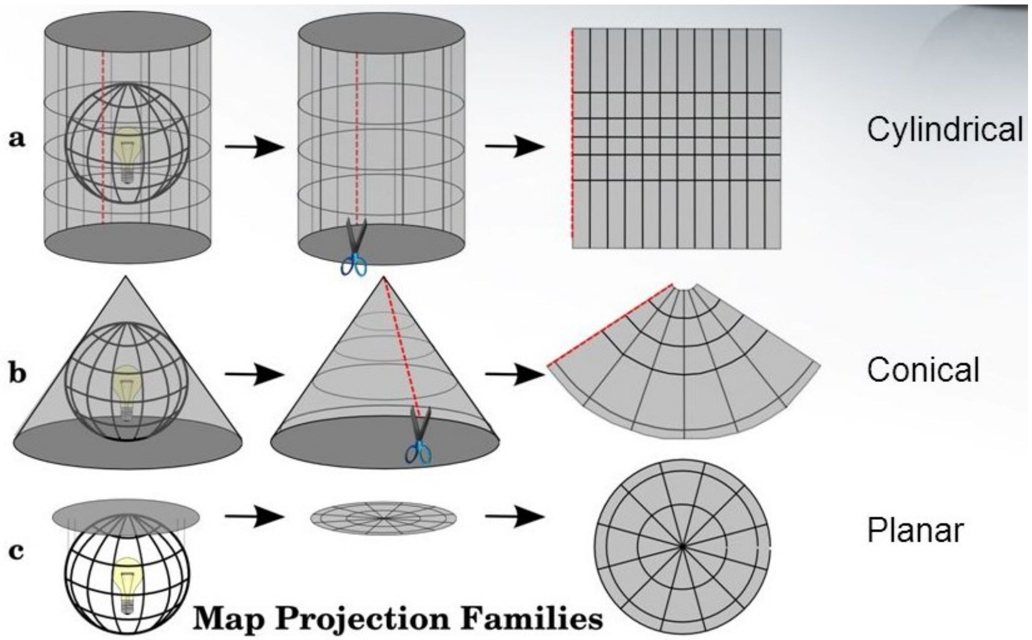
PRODUCT

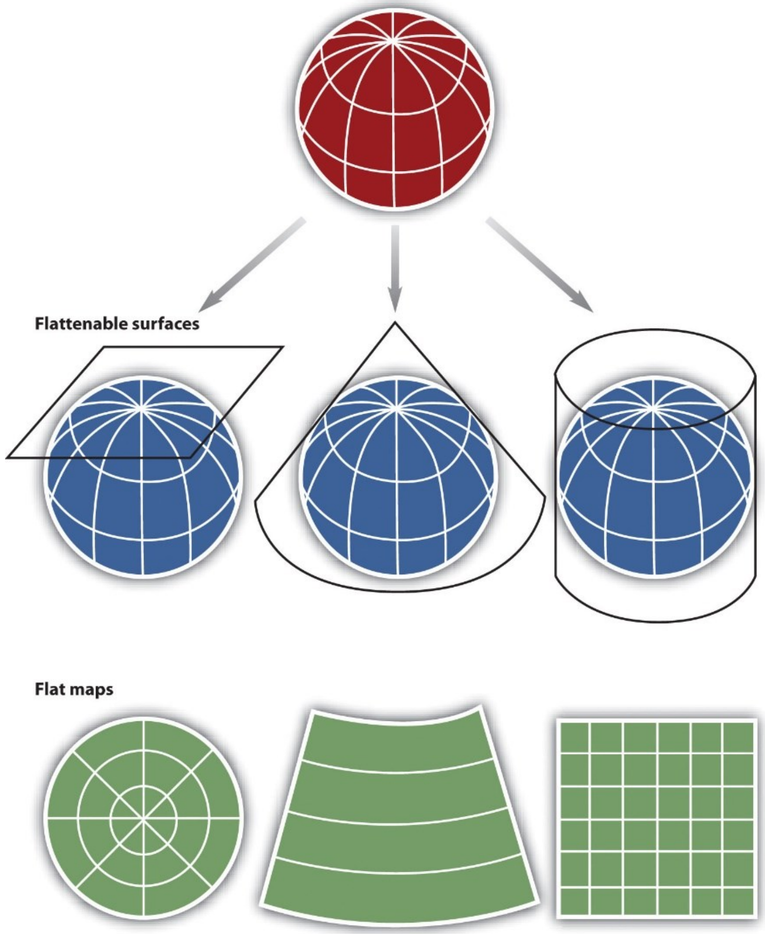
Q1:250m, 16days
A1:500m, 16days
A2:1km, 16days
A3:1km, monthly
C1:CMG, 16days
C2:CMG, monthly

Datasets per file (PROD 13A2)

```
 0) INT16 [1200x1200] 1 km 16 days NDVI
 1) INT16 [1200x1200] 1 km 16 days EVI
 2) UINT16 [1200x1200] 1 km 16 days VI Quality
 3) INT16 [1200x1200] 1 km 16 days red reflectance
 4) INT16 [1200x1200] 1 km 16 days NIR reflectance
 5) INT16 [1200x1200] 1 km 16 days blue reflectance
 6) INT16 [1200x1200] 1 km 16 days MIR reflectance
 7) INT16 [1200x1200] 1 km 16 days view zenith angle
 8) INT16 [1200x1200] 1 km 16 days sun zenith angle
 9) INT16 [1200x1200] 1 km 16 days relative azimuth angle
10) INT16 [1200x1200] 1 km 16 days composite day of the year
11) INT8 [1200x1200] 1 km 16 days pixel reliability
```

Projection & Coordinate systems





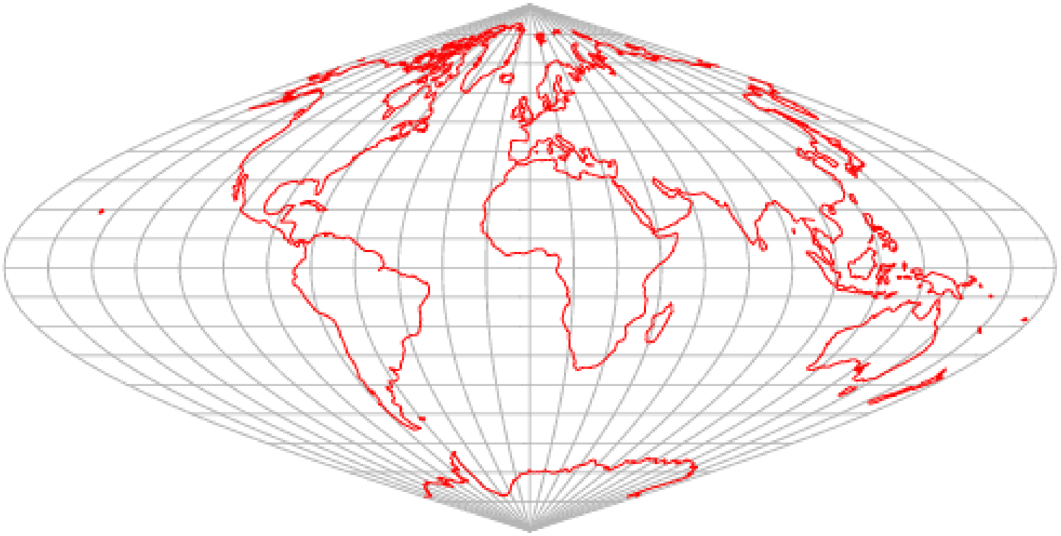
What is Geographic Projection

- The sinusoidal projection is an equal-area projection given by the transformation

$$x = (\lambda - \lambda_0)\cos\phi$$
$$y = \phi.$$

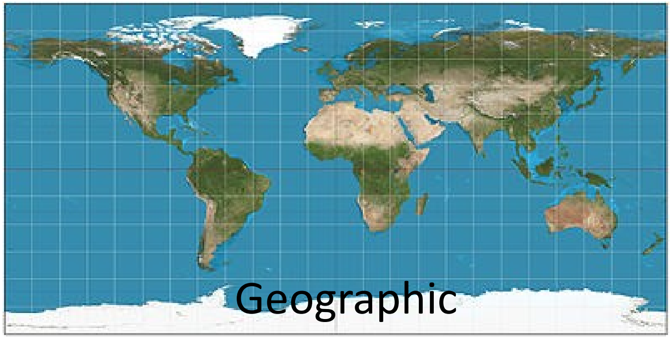
- The inverse formulas are

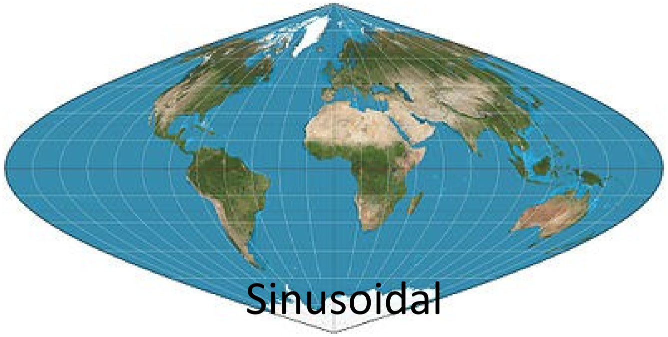
$$\phi = y$$
$$\lambda = \lambda_0 + \frac{x}{\cos\phi}.$$



Exercise #1: Reprojection

- **Read an Image**
  - Know the input projection type (Geographic in this case)
- **Convert to a new projection**
  - Reproject from **Lat/Lon** to **Sinusoidal** projection
- **Homework:**
  - Can you create the reverse process (from Sinusoidal to Geographic)
  - **Tip**: Always start by generating the Latitude and Longitude of each pixel
  - Can you estimate the pixel resolution? Explain & details?



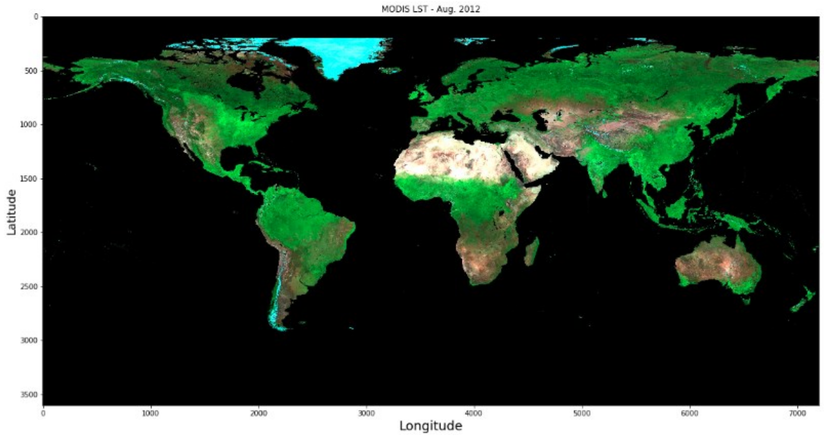


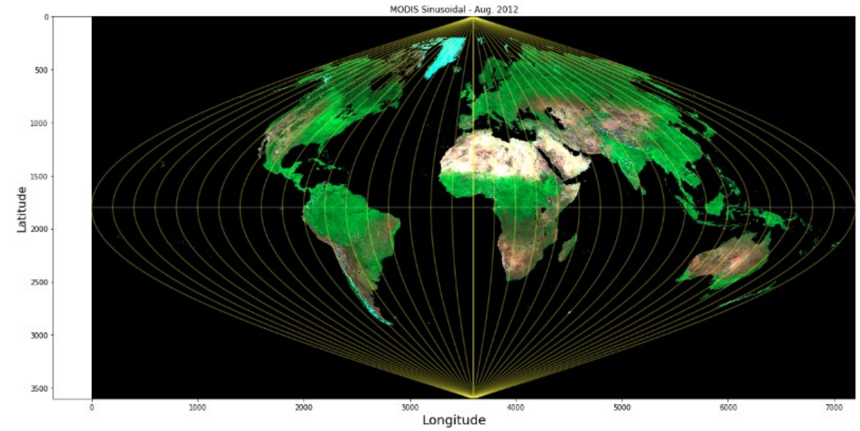
Instructions:

- Download files:
  - MODIS_LSR_August_2012.jpg and AQUA.A2016183.006.NDVI.png
  - Library: viplab_lib5.py
  - Python script: BE485_Lab10_Ex1.ipynb

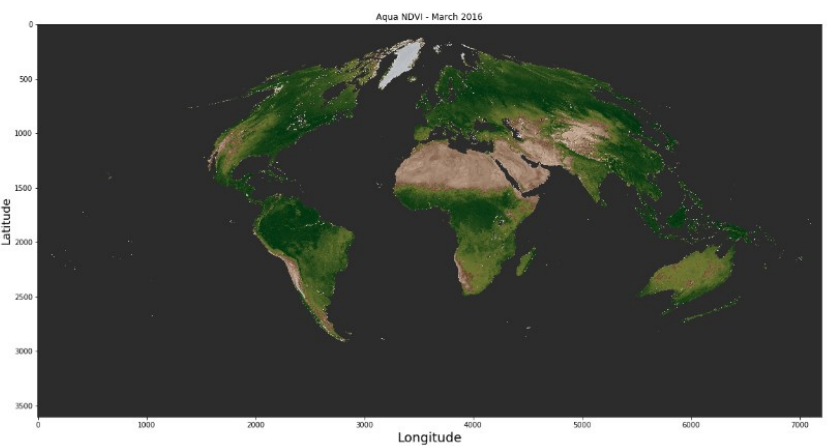
Result : Ex1

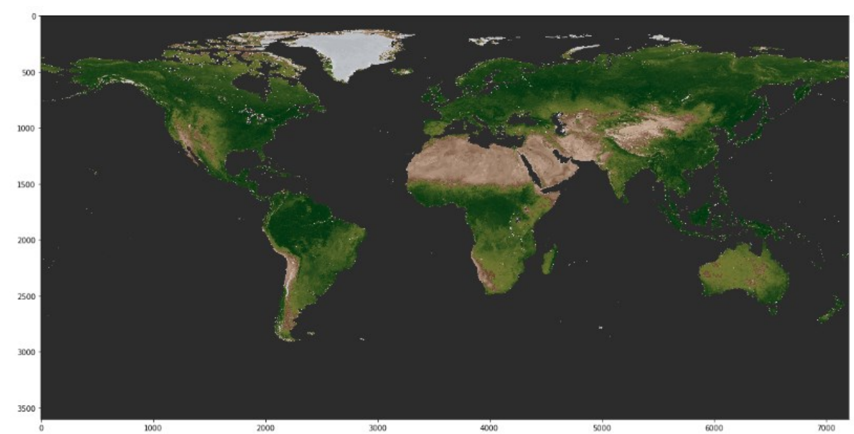
- ***Part A***: From Lat/Lon to Sinusoidal (***already done***)





- ***Part B***: From Sinusoidal to Lat/Lon (**your work**)





HDF (Hierarchical Data Format)

https://www.hdfgroup.org/

- MODIS Terra and Aqua datasets are distributed as HDF4
- Python libraries to read and write HDF files
- If you do not have them already acquire these libraries from the ***Anaconda prompt***
  - ***conda*** install hdf4
  - ***conda*** install -c conda-forge pyhdf
  - You could also do so from within JyputerLab
    - ***!conda*** install hdf4
    - ***!conda*** install -c conda-forge pyhdf

Exercise #2: Resampling

Case Location 1:

[1 km 16 days NDVI]

	601	602	603	604	605	606	607
364	3127	2972	3827	3676	3818	4340	3586
365	3247	4657	2151	3158	4144	3943	1025
366	3891	935	3479	3867	4846	1152	4837
367	3797	5670	2740	2462	1358	5714	1025
368	2357	2369	6577	6080	1262	1509	1626
369	2480	1413	4803	4415	5453	5981	1617
370	5579	5516	5735	4415	2770	5350	1674

[1 km 16 days pixel reliability]

	601	602	603	604	605	606	607
364	1	1	2	1	1	1	1
365	1	1	2	1	1	1	2
366	1	2	1	1	1	2	1
367	1	1	2	2	2	1	2
368	2	2	1	1	2	2	2
369	2	2	1	1	1	1	2
370	1	1	1	1	1	1	2

Row_1km=367, Col_1km=604
Row_3km=122, Col_3km=201

```
The pixel values for band [NDVI] at row= 122 , col= 201  are:
Input pixel= 2462
Pixel NN= 2462.0
Pixel Bilinear= 3137.0
Pixel Cubic= 3458.68
Pixel Majority= 1262.0

The pixel values for band [RANK] at row= 122 , col= 201  are:
Input pixel= 2
Pixel NN= 2.0
Pixel Bilinear= 1.5
Pixel Cubic= 1.36
Pixel Majority= 1.0
```

Case Location 2:

[1 km 16 days NDVI]

	763	764	765	766	767	768	769
190	269	771	1460	1746	3809	2957	2975
191	839	648	2360	2616	2186	2447	2421
192	169	2387	2256	2053	2041	26	-8
193	2504	2000	2055	2056	55	15	87
194	2375	2331	2241	6	125	232	2185
195	2542	2149	2144	2345	-53	2389	2488
196	2335	2198	2228	2186	169	307	4340

[1 km 16 days pixel reliability]

	763	764	765	766	767	768	769
190	2	2	2	1	1	1	1
191	2	2	1	1	1	1	1
192	2	1	1	0	0	2	2
193	1	0	0	0	2	2	2
194	1	1	1	2	2	2	1
195	1	1	1	1	2	1	1
196	1	1	1	1	2	2	0

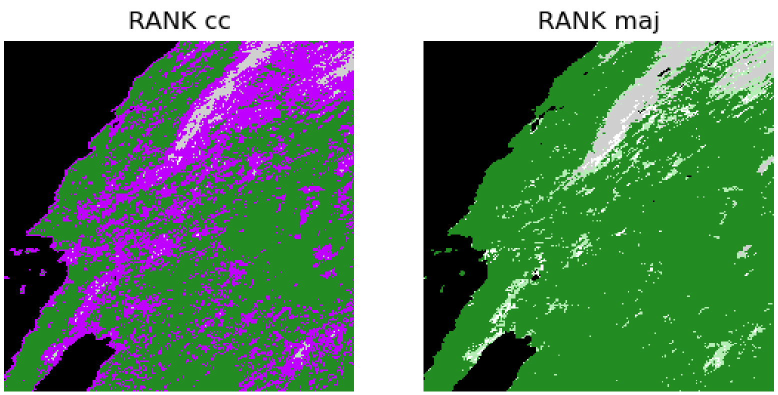
Row_1km=193, Col_1km=766
Row_3km=64, Col_3km=255

```
The pixel values for band [NDVI] at row= 64 , col= 255  are:
Input pixel= 2056
Pixel NN= 2056.0
Pixel Bilinear= 2105.0
Pixel Cubic= 1654.32
Pixel Majority= 6.0

The pixel values for band [RANK] at row= 64 , col= 255  are:
Input pixel= 0
Pixel NN= 0.0
Pixel Bilinear= 0.25
Pixel Cubic= 1.16
Pixel Majority= 0.0
```

Exercise #2: Resampling with NN, Bilinear, CC, Majority, etc.

- Read a TERRA MODIS MOD13A2 (1km) VI data file
  - NDVI (Continuous dataset)
  - RANK (Discrete, or thematic dataset)
- Apply resampling (from 1km to 3km)
  - Nearest neighbor (NN)
  - Bilinear (BI)
  - Cubic convolution (CC)
  - Majority (dominant)
- Notice the differences in the resulting image/pixel values
- Display image layers
  - Use of custom color scale to display image
  - Modify the color scale to improve the image display

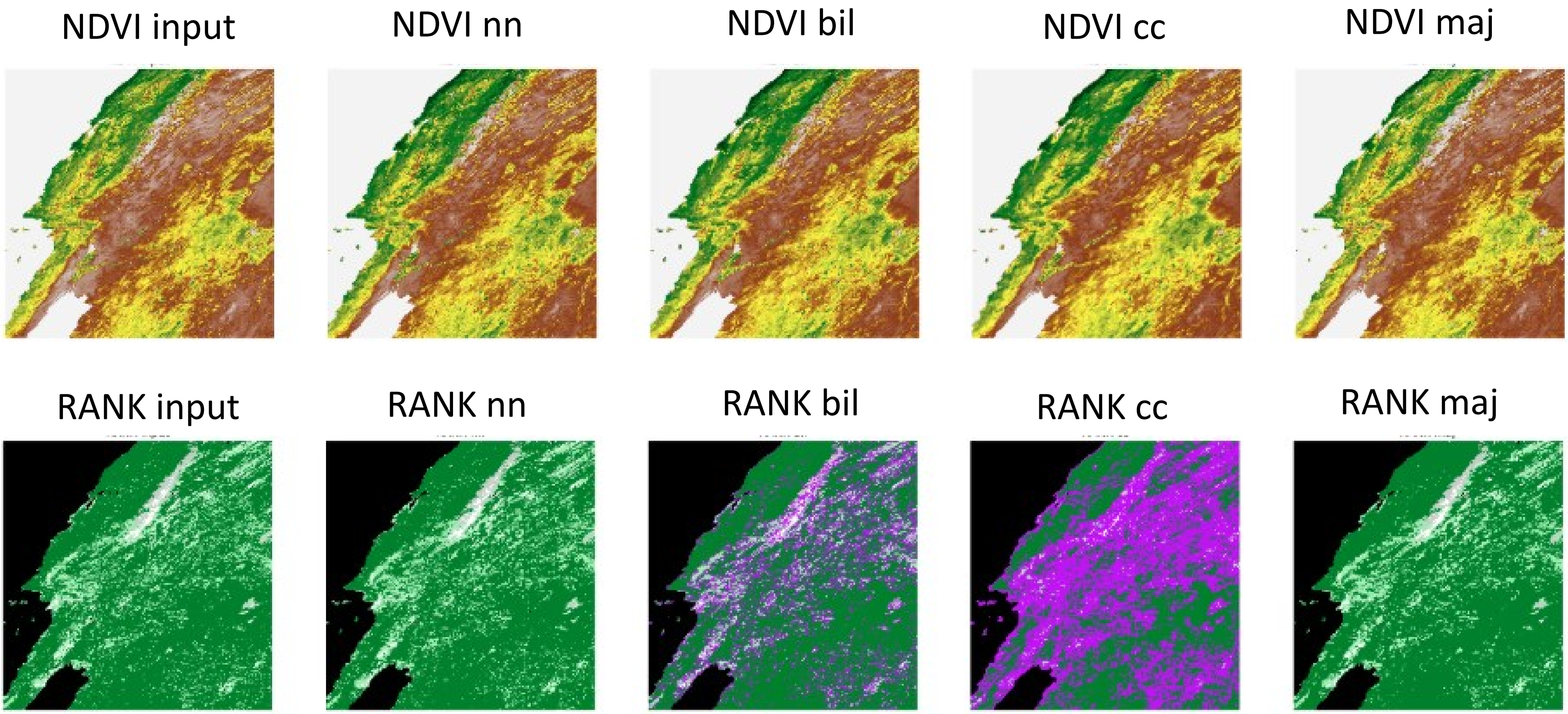


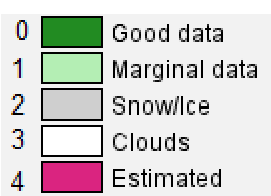
- **Homework**
  - Now apply the 4 different resampling methods to the layer 'Composite Day of the Year' (**CDOY**)
    - Read the layer that corresponds to the CDOY
  - Recall CDOY is a discrete value and evaluate the result of each resampling method
  - Now extract and plot NDVI, Rank, and CDOY, along any transect (Tip: use array slicing)
  - Display an MIR-NIR-RED false color composite image

Instructions:

- Download files:
  - Data file: MOD13A2.A2020065.h08v05.006.2020082012602.hdf
  - Library: viplab_lib5.py
  - Script: BE485_Lab10_Ex2.ipynb

Different Resampling Methods = Different Results





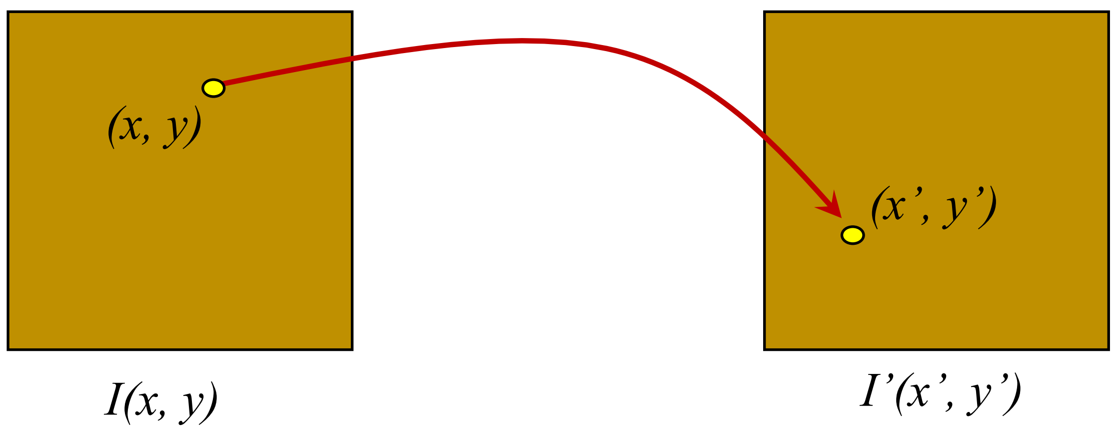
These are the only classes of pixel Rank (quality)

Depending on the resampling method the result may contain new classes that do not exist (**Magenta**) and that is an issue.

Geometric Transformation

- An operation that changes the location of a pixel/observation
- $x \rightarrow f_x(x, y) = x'$
- $y \rightarrow f_y(x, y) = y'$

$$I(x, y) = I'(f_x(x, y), f_y(x, y))$$



- **Example: Translation**

  $x' = f_x(x, y) = x+3$

  $y' = f_y(x, y) = y-1$

Forward Mapping

- Forward mapping : Source to Target

  $x \rightarrow f_x(x, y) = x'$

  $y \rightarrow f_y(x, y) = y'$

- **Problems with forward mapping**
  - ***Potential holes (target pixels with no values)***
  - ***Overlap (target pixels assigned multiple values)***

Inverse Mapping

- Inverse mapping : Target to Source

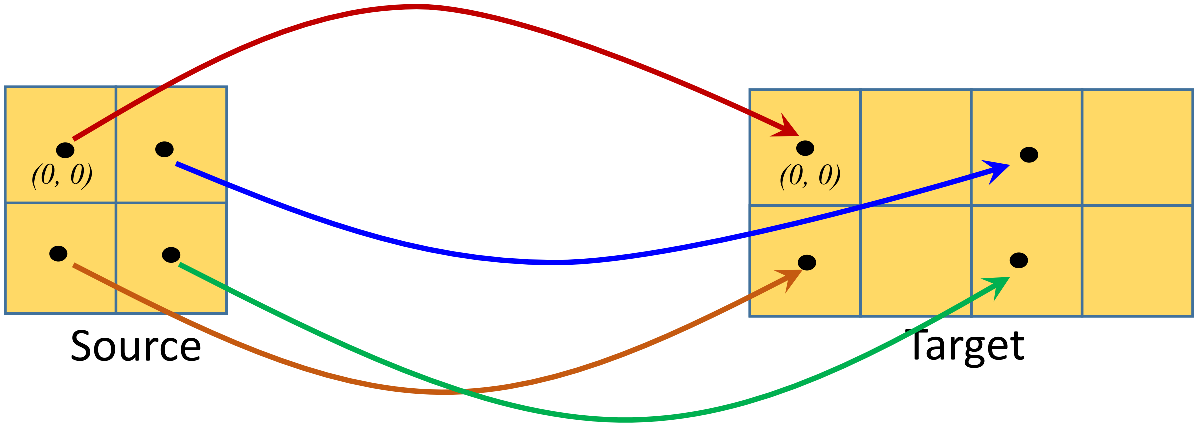
  $x' \rightarrow f_x^{-1}(x', y') = x$

  $y' \rightarrow f_y^{-1}(x', y') = y$

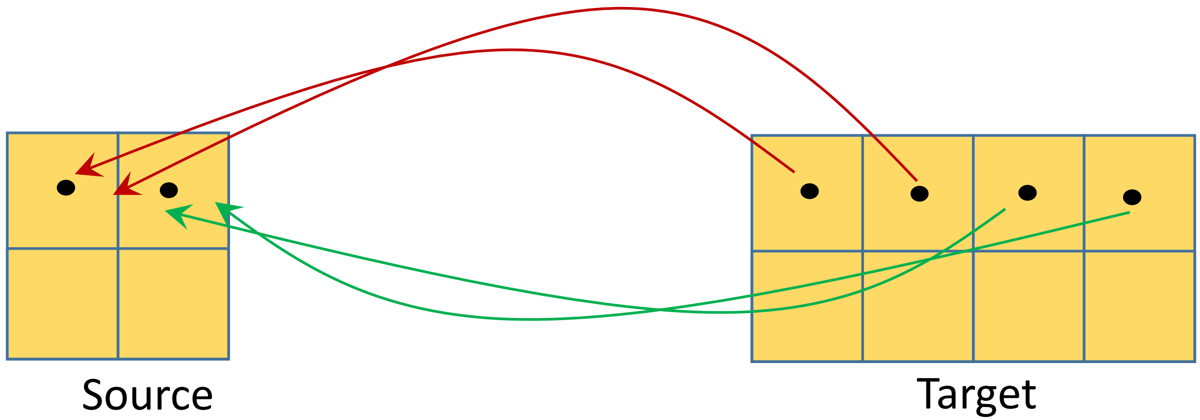
- ***Each target pixel is assigned a single value***
- ***Interpolation is required***

Examples – Scaling Transformation (along x)

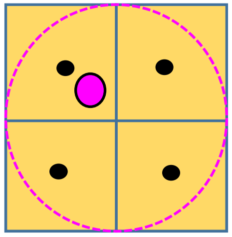
- **Forward Mapping :**
  - $x' = 2x$ *and* $y' = y$



- **Inverse Mapping**
  - $y = y'$ *and* $x = x'/2$

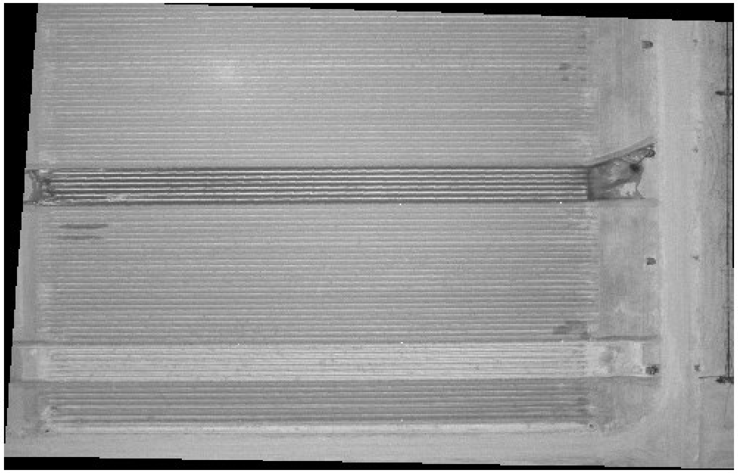


- **Interpolation:**
  - What happens when a mapping function calculates a fractional pixel address?
  - Generates a new pixel by analyzing the surrounding pixels



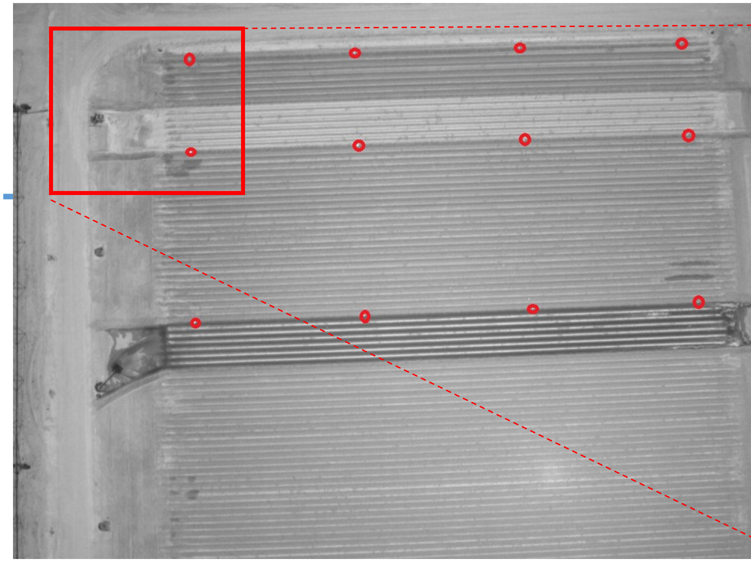
Exercise #3: Geolocation/Registration

- Read Drone Image BSQ File
- Use Ground Control Points (GCP) to geolocate/register the image
- Estimate the required coefficients for the transformation
  - Forward mapping
  - Inverse mapping
- Project Image into Map
- **Homework**:
  - Display the GCP locations in the input image
  - Save image to file as jpg/png
  - Try to load the image in Google Earth as an overlay by specifying the lat x lon box (output by your program)
    - Was the image properly Geolocated?



Instructions:

- Download files:
  - Data file: DRONE_MARICOPA_IMAGE.bsq
  - Library: viplab_lib5.py
  - Python script: BE485_Lab10_Ex3.ipynb

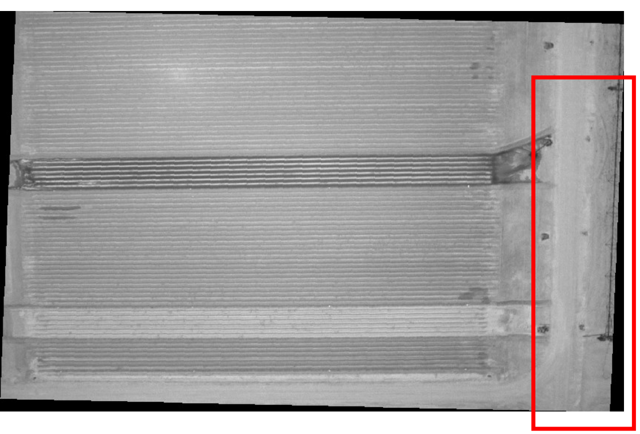


Drone Image from Maricopa Ag Center

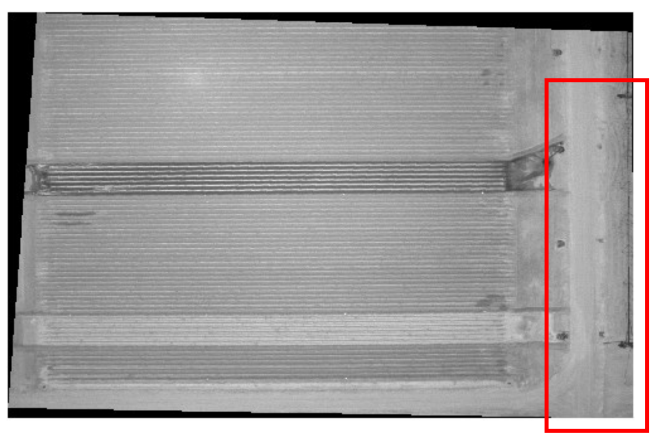
Ground Control Points (GCP)

Image		Map	
col	row	Lon	Lat
306	97	-111.9722845	33.06683512
590	86	-111.9725523	33.06683303
875	77	-111.97282	33.06683094
1154	70	-111.9730878	33.06682885
307	257	-111.9722859	33.06696341
596	245	-111.9725537	33.06696132
883	236	-111.9728215	33.06695923
1166	227	-111.9730892	33.06695714
314	550	-111.9722884	33.06719249
607	539	-111.9725562	33.0671904
898	527	-111.972824	33.06718831
1184	516	-111.9730918	33.06718622

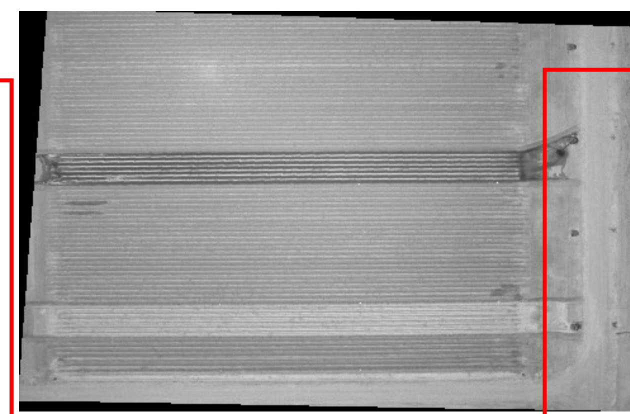
$I = f(c_0+c_1*x+c_2*y)$



$I = f(c_0+c_1*x+c_2*y+c_3*x*y)$



$I = f(c_0+c_1*x+c_2*y+c_3*x*y+c_4*x^2+c_5*y^{2)}$



- Try/Change ‘deg=’ to 3, 4, or 6 in the code