Lab 2 – Image Preprocessing and Visualization

Introduction

Pre-processing operations such as image restoration and rectification are intended to correct for sensor- and platform-specific radiometric and geometric distortions of data. Radiometric corrections may be necessary due to variations in scene illumination and viewing geometry, atmospheric conditions, or sensor noise and response. Each of these will vary depending on the specific sensor and platform used to acquire the data and the conditions during data acquisition. This lab will show some examples of how preprocessing operations can be done in Google Earth Engine (GEE). It will cover image reprojection, image registration, shadow removal, cloud removal and spectral index calculation.

Pre-lab requirement: i.e. you must complete this before lab starts

Read one of the following papers in preparation for discussion in class:

Besheer, M., Abdelhafiz, A., 2015. ***Modified invariant colour model for shadow detection.*** Int. J. Remote Sens. 36, 6214–6223. https://doi.org/10.1080/01431161.2015.1112930.

Mostafa, Y., Abdelhafiz, A., 2017. ***Accurate shadow detection from high-resolution satellite images.*** IEEE Geoscience and Remote Sensing Letters. 14, 494–498. https://doi.org/10.1109/LGRS.2017.2650996.

Scaramuzza, P.L., Bouchard, M.A., Dwyer, J.L., 2012. ***Development of the Landsat data continuity mission cloud-cover assessment algorithms.*** IEEE Trans. Geosci. Remote Sens. 50, 1140–1154. https://doi.org/10.1109/TGRS.2011.2164087.

Zitová, B., Flusser, J., 2003. ***Image registration methods: A survey.*** Image Vis. Comput. 21, 977–1000. https://doi.org/10.1016/S0262-8856(03)00137-9.

Review ***image visualization parameters*** in Lab 1.

Objectives

This lab introduces several image preprocessing techniques. At the end of this lab you will be able to use GEE to perform the following tasks:

- Extract image projection information and reproject an image.
- Register images.

- Remove shadows.
- Remove clouds.
- Compute spectral indices.

Image Projections

Projections are mathematical transformations used to take spheroidal coordinates (latitude and longitude) and transform them to a planar coordinate system. Projections always involve compromise, but careful selection of projection parameters enables creation of a map that accurately shows distances, areas, or directions. GEE is designed so that it is not typically necessary to change map projections when doing computations. Rather, GEE automatically requests inputs in the desired output projection. However, there may be instances where there is some ambiguity, e.g. if an image contains bands that do not share a common projection, or you want to use a particular projection, e.g. one that minimizes distortion of direction or distance. In such cases the reproject function can be used to modify the projection of a raster dataset, mosaic dataset, or raster item in a mosaic dataset.

Finding image projection information

```
//Find projection information for a Landsat image
// Point to a particular Landsat 5 image scene
var image = ee.Image('LANDSAT/LT05/C01/T1_TOA/LT05_015030_20100531');

//Print the projection information for the selected image to your Console tab
print('Projection and transformation information:', image.projection());

/*Note: Click on the zippy ▶ in the Console to expand and view projection details such as the
Coordinate Reference System (CRS) and the transformation parameters. Confirm that the
Landsat image uses the GEE default, i.e. the Google Mercator (EPSG:32618) projection */

//Print the nominal pixel size of the image (in meters) at the lowest level of the image pyramid
print('Pixel size in meters:', image.projection().nominalScale());
//Confirm that the nominal pixel size for the Landsat image is 30m
```

Reprojecting images

```
// Point to a MODIS vegetation index image product
var image = ee.Image('MODIS/MOD13A1/MOD13A1_005_2014_05_09');

//Print the projection information for the MODIS image to the Console
print(image.projection()); //Confirm that the CRS is SR-ORG:6974 (a sinusoidal projection)

//Display the image
var visParams = {min: 0.15, max: 0.7}; //Define visualization parameters
Map.setCenter(-77.5127,43.2642,11); //Zoom to location near Rochester at level 11
Map.addLayer(image, visParams, 'original'); //Add the original image to the map.

//The pixels in the image look somewhat distorted, but can be reprojected to Maps Mercator
var reprojected = image
```

```
   .reproject('EPSG:4326'); //EPSG:4326 is the code for ellipsoidal coordinates based on WGS 84
Map.addLayer(reprojected, visParams, 'Reprojected');
```

Performing image registration

Image registration is the process of transforming different sets of data a common spatial reference system. Data may come from different sensors, times, depths, or viewpoints and comparison requires bringing them to a common georeferencing system.

```
/* There are two steps to registering an image in GEE: determining a displacement image, which
contains bands dx and dy that show an offset in x and y, respectively for each pixel, and then
applying this to the image that is to be registered.  This can be performed as two separate steps
(Approach 1 below) or as a single step (Approach 2).  Either way, the registration is using rubber
sheeting. */

// Load two images to be registered.
var image1 = ee.Image('SKYSAT/GEN-A/PUBLIC/ORTHO/MULTISPECTRAL/s01_20150502T082736Z');
var image2 = ee.Image('SKYSAT/GEN-A/PUBLIC/ORTHO/MULTISPECTRAL/s01_20150305T081019Z');

//Resampling in GEE can use nearest neighbor, bilinear or bicubic interpolation
var image1Orig = image1.resample('bicubic'); //This forces the resampling method to bicubic
var image2Orig = image2.resample('bicubic'); //Without using resample, the default is nearest neighbor

//Choose to register using only the 'R' band.
var image1RedBand = image1Orig.select('R'); //Create a new image that contains only the red band
var image2RedBand = image2Orig.select('R');

//Approach 1

//Determine the displacement needed to register image2 to image1 (using the selected red bands).
var displacement = image2RedBand.displacement({
  referenceImage: image1RedBand,
  maxOffset: 50.0,      //defines the maximum displacement between the two images
  patchWidth: 100.0   //size of patch used to determine image offsets.
});

// Use the computed displacement to register all original bands.
var registered = image2Orig.displace(displacement);

// Show the results of co-registering the images.
var visParams = {bands: ['R', 'G', 'B'], max: 4000};
Map.addLayer(image1Orig, visParams, 'Reference');
Map.addLayer(image2Orig, visParams, 'Before Registration');
Map.addLayer(registered, visParams, 'After Registration');

//Approach 2
/*If you don't need to know what the displacement is, Earth Engine provides the register()
method, which combines calls to displacement() and displace(). The register() function uses all
bands of the image in performing the registration. */
var alsoRegistered = image2Orig.register({
```

```
  referenceImage: image1Orig,
  maxOffset: 50.0,
  patchWidth: 100.0
});
Map.addLayer(alsoRegistered, visParams, 'Also Registered');
```

Detect and remove shadow pixels (2 approaches)

This section will guide you through two approaches that detect and removes shadow pixels using different color models and thresholding methods. There are various color models or spaces, such as Red-Green-Glue (RGB), Hue-Saturation-Value (HSV), $C_1C_2C_3$ and others. **Approach 1** below utilizes the C_3^* index, which is originally based on the $C_1C_2C_3$ color model. Besher and Abdelhafiz (2015) found that the C_3 component was sensitive to shadows but was not stable for certain color values. Thus, the C_3^* index was introduced to include near-infrared (NIR) information to increase the stability of the original C_3 index. A threshold value is then used to distinguish between shadow and non-shadow pixels based on a histogram of C_3^* values. If water features are present in the image, a water mask is also applied since water and shadow pixels have similar radiometric responses. **Approach 2** utilizes the HSV color space and applies the normalized saturation-value difference index (NSVDI). A threshold value is used to distinguish between shadow and non-shadow based on a histogram of NSDVI values. Mostafa and Abdelhafiz (2017) describe these two approaches as the best approaches in shadow detection due to the use of direct formula, and an automatic procedure that does not require human input.

```
/*Define study boundary and name it as geometry. This is a necessary step to limit the study
boundary to the image of interest.*/
var geometry = ee.Geometry.Polygon(
      [[[-76.1397, 43.0399],
        [-76.1399, 43.0314],
        [-76.1283, 43.0314],
        [-76.1284, 43.0397]]]);

//Create a variable called NAIP to point to the NAIP data collection
var NAIP = ee.ImageCollection('USDA/NAIP/DOQQ');

//Filter NAIP imagery
var point = ee.Geometry.Point(-76.136, 43.036); //Create a point at City of Syracuse
var naipImage = NAIP
  .filterBounds(point) //filter with defined filter location
  .filterDate('2011-01-01','2011-12-30') //Limit images to the ones collected in 2010
  .mosaic(); //Convert selected images into one composite image rather than an image collection.

Map.setCenter(-76.1362, 43.0362, 15);
Map.addLayer(naipImage,{},'original NAIP image');
```

Approach 1: Modified C_3^* Index (from Besheer and Abdelhafiz, 2015).

```
//Construct an image with green, red and near infrared bands of selected NAIP image
var imageGRN = naipImage.select(['G','R','N']);

/*Spatial reducers are functions in GEE that composite all the images in an Image Collection to a
single image representing, for example, the min, max, mean or standard deviation of the images.
It can also be used to composite the maximum value per each pixel across all bands in one image.
Here, it reduce the GRN image to a one-band image with the maximum DN value for each pixel
across all bands in the GRN image.*/
var maxValue = imageGRN.reduce(ee.Reducer.max());

// Merge the one-band maximum value image and the original NAIP image to create a new image.
var imageMAX = naipImage.addBands(maxValue.select(['max']),["max"]); /*first max selects the
 band to add, second max provides the name of the band in the new image*/

// Calculate C3* index (from Besheer and Abdelhafiz, 2015)
var C3 = imageMAX.expression(
   'atan(B/max)', {
     'B':image.select('B'),
     'max':image.select('max')
   });

/* Print a histogram of the C3* index and determine the inflection point. Besheer and Abdelhafiz
(2015) found experimentally that selecting the low frequency DN in the valley between the
predominant features gave consistently accurate threshold levels for separating the shadow from
non-shadow regions. See Figure 2 in Besheer and Abdelhafiz (2015) for more details. */
print(ui.Chart.image.histogram(C3,geometry,1));

//Based on selected threshold above, mask out non-shadow from the C3 image
var shadowmask = C3.select('B').gte(0.85); //create non-shadow mask based on C3 threshold
var C3shadow = C3.updateMask(shadowmask); //apply mask C3 image to get shadow-only image

// Apply a NDWI mask to the shadow image to mitigate confusion between water and shadows.
// Generate a NDWI image based on the selected NAIP image bands
var NDWI = imageMAX.expression(
   '(G-N)/(G+N)', {
     'G':image.select('G'),
     'N':image.select('N')
   });

// Print a histogram of the NDWI values and determine low point in the last valley.
print(ui.Chart.image.histogram(NDWI,geometry,1));
```

```
// Based on the threshold selected above, mask out water pixels from the shadow-only C3 image
var NDWImask = NDWI.select('G').lte(0.6);  //create a water mask based on selected threshold in step above
var C3shadow = C3shadow.updateMask(NDWImask); //apply defined mask from above to the shadow-only C3 index image

//Display final shadow pixels with water removed. This sets the stage for shadow compensation, which is the next key step in shadow detection and removal. Shadow compensation can be done by applying equation 17 in Mostafa and Abdelhafiz (2017).
Map.addLayer(C3shadow, {palette: 'FF0000'}, 'shadow_final');
```

Approach 2: Normalized saturation-value difference index (NSVDI) (from Mostafa and Abdelhafiz, 2017).

```
/* Extract RGB bands from the selected NAIP and divide all bands by 255 to convert original DN values (0-255) to a range of 0-1, which can be processed by the next rgbToHsv function in the next step. */
var naipImage = naipImage
            .select(['R','G','B'])
            .divide(255);

/*Convert NAIP image into Hue-Saturation-Value (HSV) space. HSV space is also referred to as IHS (intensity, hue, saturation) or HSB (hue, saturation, brightness). HSV system is an alternative to describing colors using RGB components. Value relates to the brightness of a color. Hue refers to the dominant wavelength of light contributing to the color. Saturation specifies the purity of color relative to gray. It is often utilized in operations for resolution-enhancement (Lillesand et al. 2015). */
var naipImagehsv = naipImage.rgbToHsv();

//Generate a NSVDI image based on the converted HSV image from step above
var NSVDI = naipImagehsv.expression(
   '(S-V)/(S+V)', {
    'S':naipImagehsv.select('saturation'),
    'V':naipImagehsv.select('value')
   });

/*Print a histogram of the NSVDI in order to determine the shadow threshold. Mostafa and Abdelhafiz (2017) expect the threshold to be zero, but this did not provide a satisfactory result. An iterative process was needed to identify a threshold (-0.2). */
print(ui.Chart.image.histogram(NSVDI,geometry,1));

//Based on selected threshold above, mask out non-shadow pixels from the NSVDI image
var NSVDImask = NSVDI.select('saturation').gte(-0.2); //select non-shadow pixels based
var NSVDIshadow = NSVDI.updateMask(NSVDImask); //apply mask to remove non-shadow pixels.
```

```
// Display the shadow-only NSVDI image
Map.addLayer(NSVDIshadow,{palette: 'FF0000'});
```

Cloud detection

This section explores the detection of cloud pixels utilizing the built-in Landsat cloud score function in GEE that compute a score of cloudiness for each pixel. Note that cloud detection is a first step in compensating for the impact of cloudy pixels.

```
// Load a cloudy Landsat scene and display it.
var cloudy_scene = ee.Image('LANDSAT/LT05/C01/T1_TOA/LT05_015030_20100531');
Map.centerObject(cloudy_scene);
Map.addLayer(cloudy_scene, {bands: ['B4', 'B3', 'B2'], max: 0.4}, 'TOA');

/* Add the cloud score band, which is automatically called 'cloud'. The simpleCloudScore function
 uses brightness, temperature, and NDSI to compute a score in the range [0,100]. */
var scored = ee.Algorithms.Landsat.simpleCloudScore(cloudy_scene);

/* Create a mask from the cloud score by experimentally selecting a threshold. Smaller thresholds
mean more pixels are selected as clouds. */
var mask50 = scored.select(['cloud']).lte(50); // selects pixels with cloud score of 50 or less
var mask30 = scored.select(['cloud']).lte(30);

// Apply the masks created above to the image and display the result to determine an appropriate threshold.
var masked50 = cloudy_scene.updateMask(mask50); //apply mask50
Map.addLayer(masked50, {bands: ['B4', 'B3', 'B2'], max: 0.4}, 'masked50');
var masked30 = cloudy_scene.updateMask(mask30); //apply mask30
Map.addLayer(masked30, {bands: ['B4', 'B3', 'B2'], max: 0.4}, 'masked30');
```

Performing spectral index calculation (review from lab 1)

Spectral indices combine surface reflectance at two or more wavelengths. Through careful band selection, indices can be targeted to indicate relative abundance of features of interest.

```
// Load an image.
var image = ee.Image('LANDSAT/LT05/C01/T1_TOA/LT05_015030_20100531');
```

Enhanced Vegetation Index (EVI)

```
var evi = CF.expression(
   '2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))', {
     'NIR':image.select('B4'), //select 4th band
     'RED':image.select('B3'), //select 3rd band
     'BLUE':image.select('B1') //select 1st band
});
Map.addLayer(evi);
```

Assignment – Answer the following questions

Submit your code and text answers for this assignment by clicking on "Get Link" in the Code Editor and sending the link generated to the TA (your link will look something like https://code.earthengine.google.com/XXXXXXXX. Any written responses should be in comments with each question separated by a line of forward slashes. For example:

```
//Q1. Text answer to Q1
Code answer to Q1  //All code responses need suitable comments.

////////////////////////////////////////////////////////////
//Q2. Text answer to Q2
Code to Q2
...
```

1. How do you reproject images in Google Earth Engine? Under what scenarios would you need to do so?

2. You are analyzing a series of aerial images, but you found the images are not spatially matched. Use a pair of NAIP image scenes to illustrate how would you adjust the minor mismatch between the images.

3. Using LANDSAT/LT05/C01/T1_TOA/LT05_015030_20100531 as your input, calculate four indices (see Harris Geospatial, 2018) in EE and describe the differences between them. Explain how each index might be applied.

4. What is the theory behind the bands used to calculate NDVI and EVI? How do these indices support distinguishing vegetation from other cover types? Are there any limitations or alternatives to using these indices?

5. Select one NAIP image (do not use the same scene from this lab) and use one of the shadow detection methods illustrated in this lab to identify all shadow pixels within the image. Submit code to demonstrate both original image and the image without shadow in it.

6. (Bonus) Select any satellite image series, construct an NDVI time series with minimum cloud and shadow artifacts. Hint: use the method presented by Chen et al. (2004).

Appendix A - Glossary of Terms

- **Projection**: Projections are a mathematical transformation that take spherical coordinates (latitude and longitude) and transform them to an XY (planar) coordinate system. This enables you to create a map that accurately shows distances, areas, or directions.
- **Reproject**: In GEE, the reproject function modifies the projection of a raster dataset, mosaic dataset, or raster item in a mosaic dataset.
- **Scale**: Scale defines the relationship between the size of a feature depicted on a map to its actual size.
- **Palette**: In computer graphics, a palette is a finite set of colors. Palettes can be optimized to improve image accuracy in the presence of software or hardware constraints.
- **Image registration**: Image registration is the process of transforming different sets of data a common spatial reference system. Data may be from different sensors, times, depths, or viewpoints.
- **Spectral indices**: Spectral indices are combinations of surface reflectance at two or more wavelengths. Bands are selected such that an index will highlight certain characteristics.

Appendix B – Useful Resources

Google Earth Engine

- Google Earth Engine Java Style Guide
  http://google.github.io/styleguide/javascriptguide.xml

- Google Earth Engine Guides/Cookbook/Dictionary
  https://developers.google.com/earth-engine/

- Google Earth Engine API Tutorials
  https://developers.google.com/earth-engine/tutorials

- Google Earth Engine Workshop (Beginning)
  https://docs.google.com/document/d/1ZxRKMie8dfTvBmUNOO0TFMkd7ELGWf3WjX0JvESZdOE

- Google Earth Engine Workshop (Intermediate)
  https://docs.google.com/document/d/1keJGLN-j5H5B-kQXdwy0ryx6E8j2D9KZVEUD-v9evys

- Google Earth Engine tutorials on AmericaView
  https://americaview.org/program-areas/education/google-earth-engine-tutorials/

General references

Chen, J., Jönsson, P., Tamura, M., Gu, Z., Matsushita, B., Eklundh, L., 2004. A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky-Golay filter. *Remote Sens. Environ*. 91, 332–344. doi:10.1016/j.rse.2004.03.014

Harris Geospatial, 2018. Spectral Indices. URL http://www.harrisgeospatial.com/docs/alphabeticallistspectralindices.html (accessed 25 Feb 2018).

Horning, N., 2004. Selecting the appropriate band combination for an RGB image using Landsat imagery Version 1.0. American Museum of Natural History, Center for Biodiversity and Conservation. https://www.amnh.org/content/download/74355/1391463/file/SelectingAppropriateBandCombinations_Final.pdf (accessed 25 Feb 2018)

Kennedy M., 2000. Understanding Map Projections. GIS by ESRI 1–31. http://www.icsm.gov.au/mapping/images/Understanding_Map_Projections.pdf

Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2015). Remote sensing and image interpretation. 7th edition. John Wiley & Sons.

This work was produced by Ge (Jeff) Pu and Dr. Lindi Quackenbush at State University of New York-College of Environmental Science and Forestry. Suggestions, questions and comments are all welcome and can be directed to Jeff at gpu100@syr.edu.