Dividing Rasters into Best-Fitting Polygons

By Alex Abramson

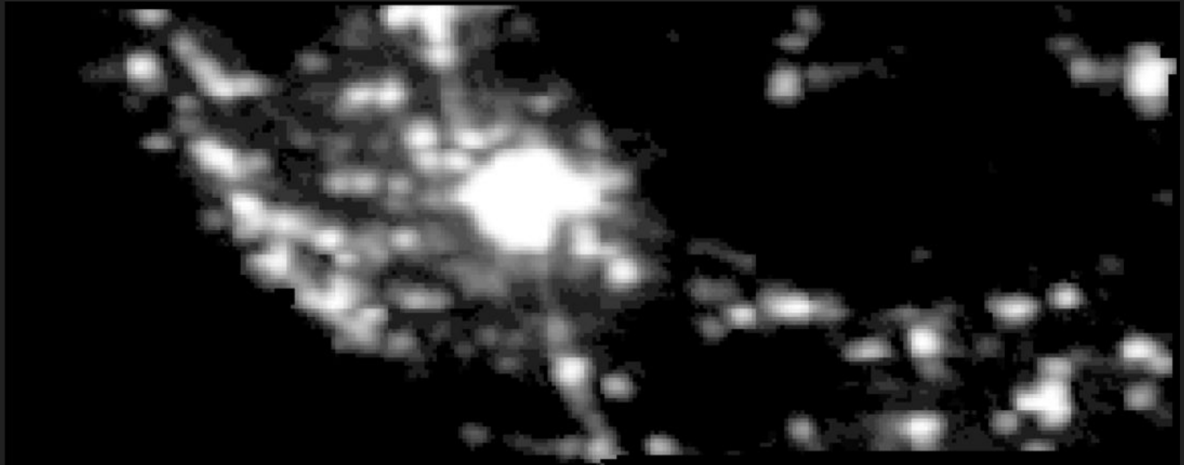
An ArcPy Project Supported By Google Earth Engine

Problem:

1. Traditional Raster to Polygon tool will, if applied to a highly varied raster, produce a feature class with many different polygons of different values. This can be hard to use for analysis.
2. Reclassifying similar values can help -- but:
   a. Which values to reclassify together can be highly subjective
   b. If we care about identifying local minima and maxima, this is problematic. The peak of one local maximum may only have the value of a ring around another higher value local maximum.
3. We need another method to divide rasters into polygons representing local maxima and minima

Applied Problem: Nightlighting in a Metro Area

Challenge: Say we have a nightlighting raster of a multi-county metro area. We want to comprehensively divide the metro area so we can perform comparative analysis on each node of light (and of darkness). Local maxima/minima represent very few cells -- we need to group these cells with nearby cells. We want to group these cells with the nearby node which they most belong to/are most similar to. But how would we group these neighboring cells? With which local maximum/minimum node? Particularly those neighboring cells which lie somewhere between nodes?



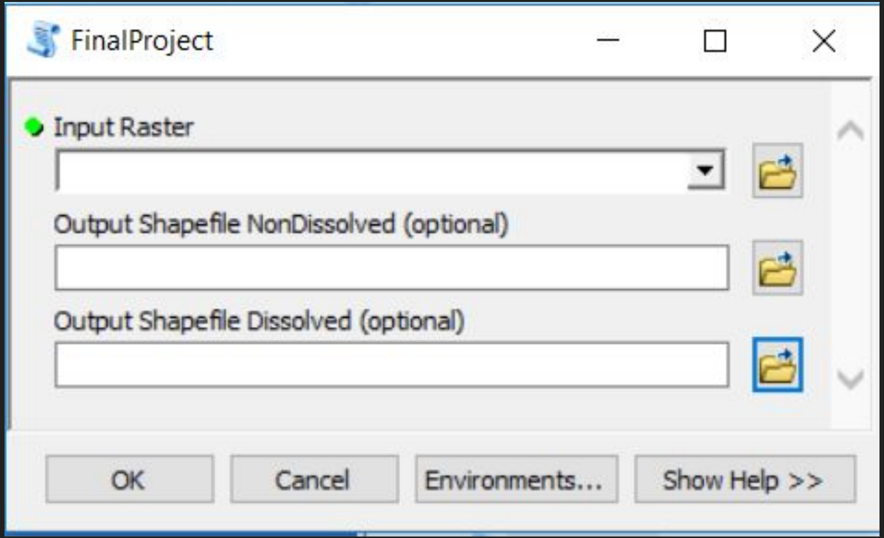
Goal: Systematically Divide Cells into Polygons Representing Nodes and their Hinterlands

This toolset allows us to:

1. Divide a raster into polygons centering on local minima/maxima and containing those cells which are more similar to that node than to other nodes
2. Subsume polygons of minor nodes within neighboring polygons of major nodes, until only a few polygons capture the big picture of the raster

Step 1: Which Node to Assign Neighbors to?

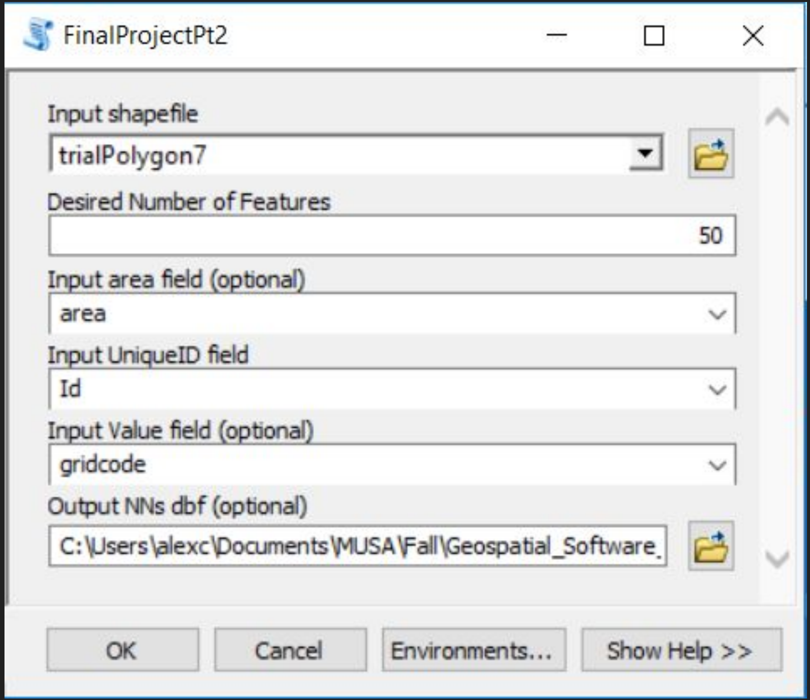
1. For each cell in raster, look at it and it's adjacent neighbors:
    a. Which neighbor has the highest value? The lowest value?
    b. Is this cell's value closest to the highest value neighbor or the lowest?
2. Change cell's value to whichever of those two cells had a value that this was closest to
3. Repeat until equilibrium achieved, i.e. further processing yields no improvement
4. Output shapefile of polygons (dissolved version optional although not recommended)
5. But unfortunately, output can be chaotic -- too many local maximums and minimums, too many polygons.



(Note that local maxima/minima cells never change their values -- within their neighbors, they are always the closest maximum or minimum to themselves)

Step 2: Which Nodes (and their Polygons) are Most Important?

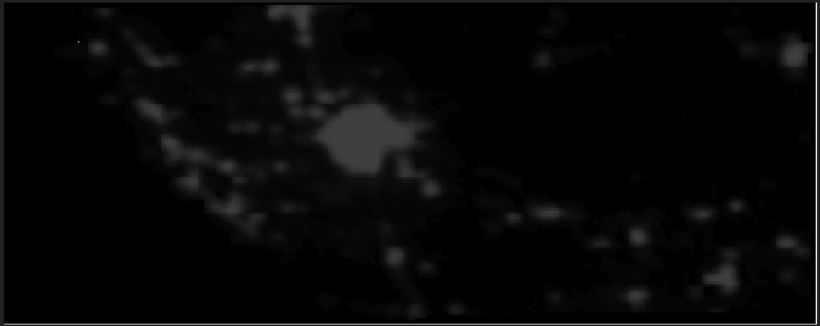
1. Identify all pairs of neighboring polygons. How big is each neighbor? What value does each have?
   a. Big neighbors should annex their smaller neighbors first
      i. (And smaller polygons should never annex their bigger neighbors)
   b. Neighbors of similar values should merge before neighbors of drastically different values do
   c. How do we weigh these goals?
      i. Create measure = (Difference in Value)^2/Difference in Area
      ii. Pairings with low measure value are best candidates for annexation
         1. E.g. a big dark area subsuming a small island of marginally less darkness
2. Combine the pair of neighbors with the lowest measure value. Give the combined polygon the value of the bigger neighbor.
3. Repeat until polygons have been culled to the desired number and output this shapefile

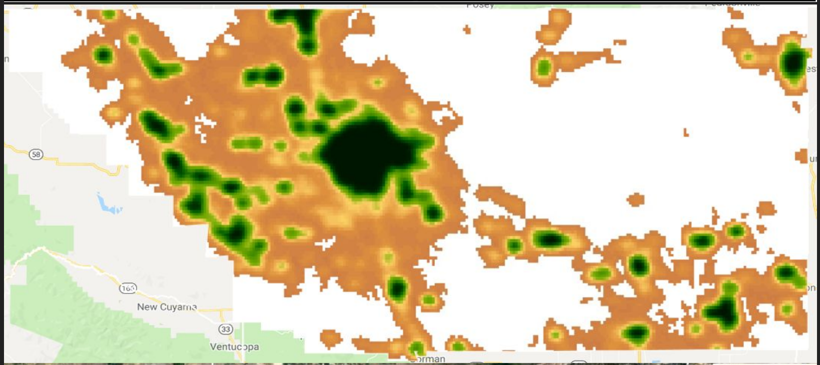


Example: Bakersfield Metro Area -- Data Prep

Obtain Nightlighting Imagery using Google Earth Engine:

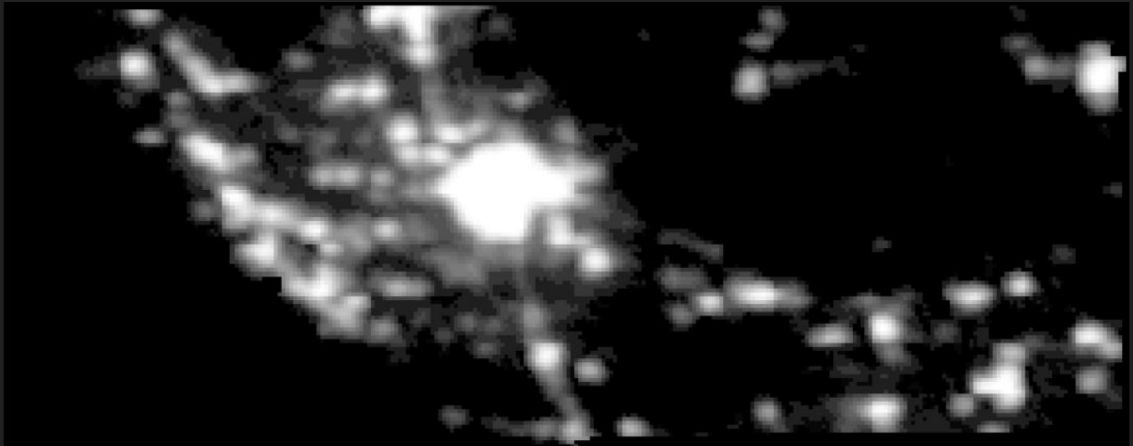
1. Import DMSP OLS: Nighttime Lights Time Series and US county boundaries
2. Combine the relevant county features as one feature collection and union them into one feature
3. Extract stable light band from Nighttime Lights
4. Clip stable light band by the metro area
5. Export clipped Nighttime Lights at the appropriate resolution as a TIF





Example: Bakersfield Metro Area -- Step 1

The first step creates polygons out of all local minima/maxima and their neighboring cells. Note that the general pattern follows the raw nightlighting raster displayed below, although the presence of noodle-shaped polygons is problematic -- these will need to be worked on later.

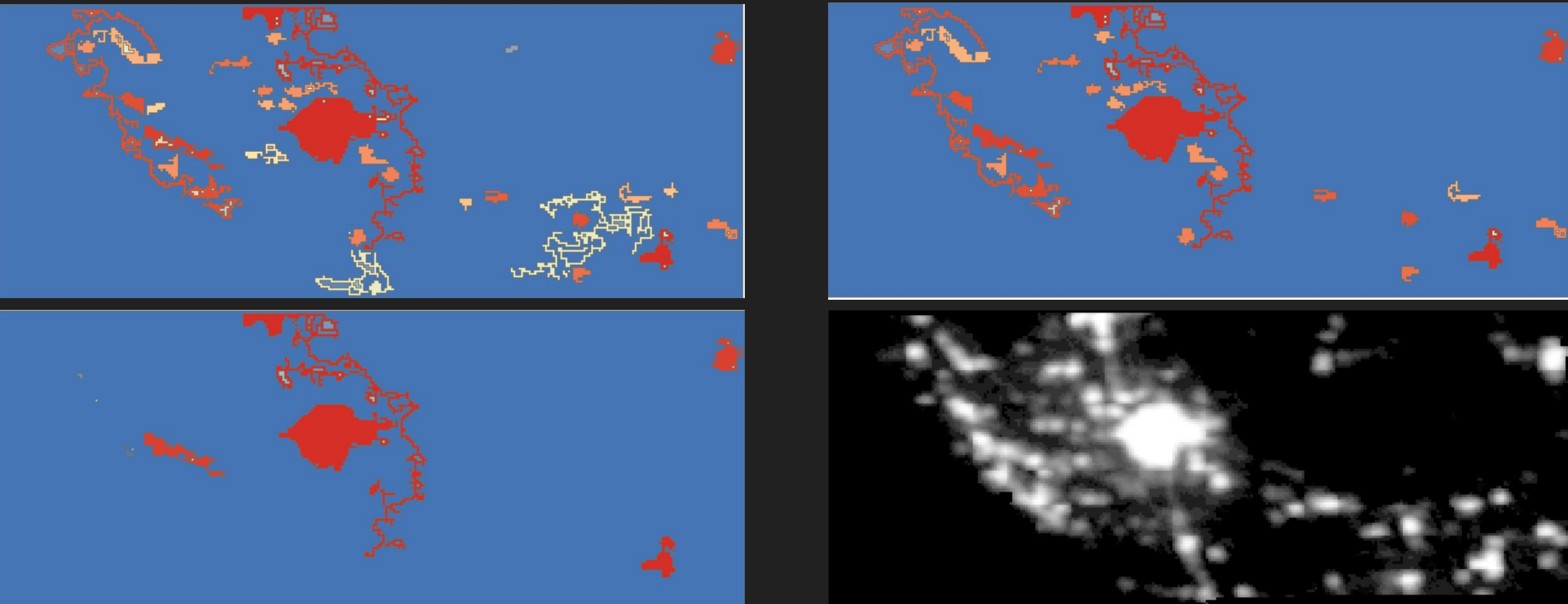


Example: Bakersfield Metro Area -- Step 2

The images below depict output polygons at varying levels of desired specificity. Note that many but not all of the noodle shaped polygons are removed. Further tinkering with the Measure will be pursued to improve the balance between the goals of area difference maximization and value difference minimization.

Top Left -- grouped into 150 Polygons
Bottom Left -- grouped into 50 Polygons

Top Right -- grouped into 100 Polygons
Bottom Right -- Actual Nightlighting



Next Steps:

Clearly there is still much work to be done here. Next steps would center on:

1. Neighborhood for focal statistics in part 1 -- should this include diagonally adjacent cells? Should it be a 5x5 neighborhood? Are maximum and minimum the right statistics to compute?
2. How is it best to weight value difference minimization and area difference maximization in prioritizing candidate pairs for annexation? Are these even the right goals to shoot for in this process?
3. In annexation process, should new feature take on all characteristics of larger constituent polygon? Or should both candidates' values be weighted in proportion to their area?