Cell Service Navigation

We will divide this problem into two sub-problems:

- Finding where the hiker is
- Finding where the hiker should go to obtain better reception

We will first find the hiker, of which we know the following information

- Is on water
- Is on flat surface
- Not in shadow
- Near forest-water boundary and mountains
- When facing a forest-water boundary/mountain, has a shadow that is perpendicular to his face during the late-morning of a Northern Maine winter

Criterion 1: On Water

We know for sure that the hiker is on water. Thus, we simply need a true/false map of water surfaces to eliminate any surfaces that is not water. Use RECLASSIFY on *SurfaceWater*.

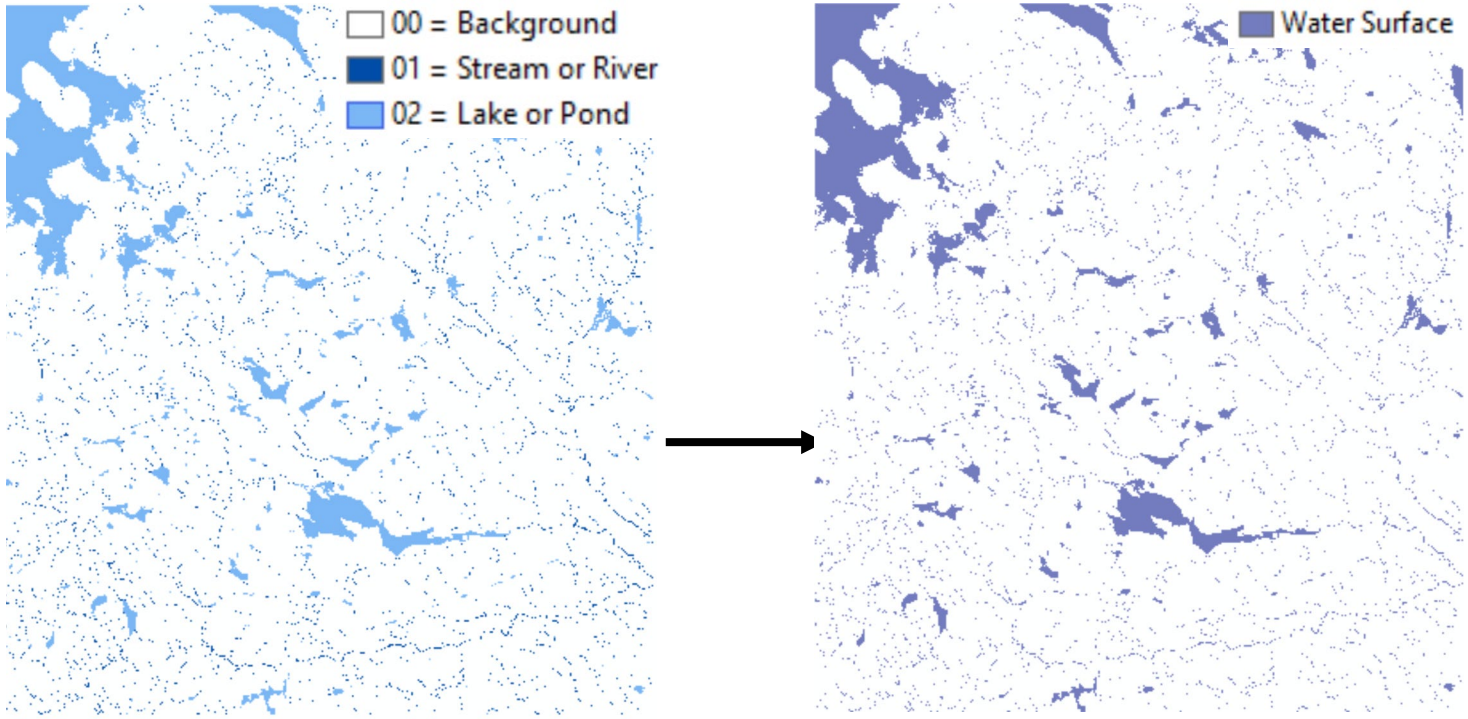


Figure 1a. SurfaceWater
Description: water surfaces
Source: provided in sample

Figure 1b. water_tf
Description: true or false of water surfaces
Source: RECLASSIFY(*SurfaceWater*)

Background→ noData
Stream or River → 1
Lake or Pond → 1

Criterion 2: Flat Surface

Judging from the picture, the hiker is on a relatively flat surface. We will use SLOPE on *Elevation* to find the slope of the area. We will then RECLASSIFY to set values that are greater than a certain slope to NoData. For the remaining slopes, we will RECLASSIFY and give them a score from 1 – 5, with 5 given to surfaces with the lowest slopes

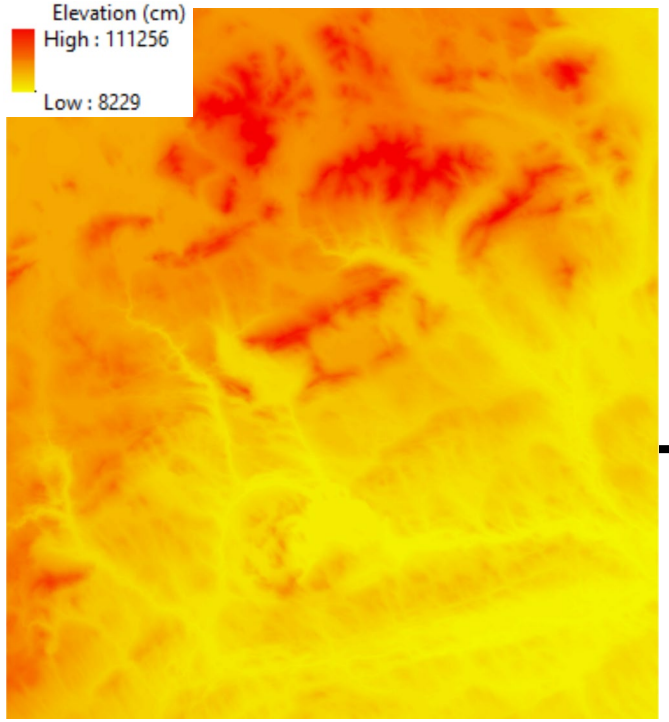


Figure 2a. elevation

Description: elevation
Source: provided in sample

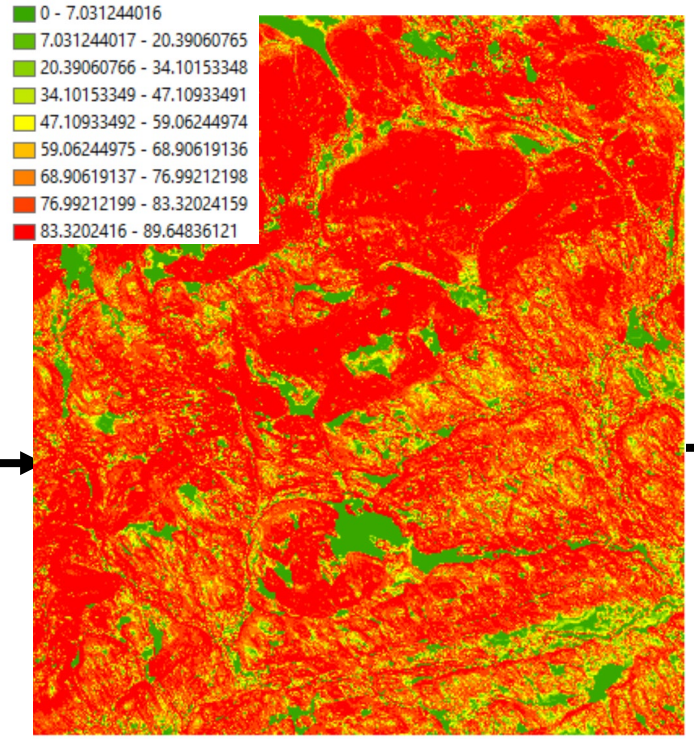


Figure 2b. slope

Description: slope of surface
Source: SLOPE(*elevation*)

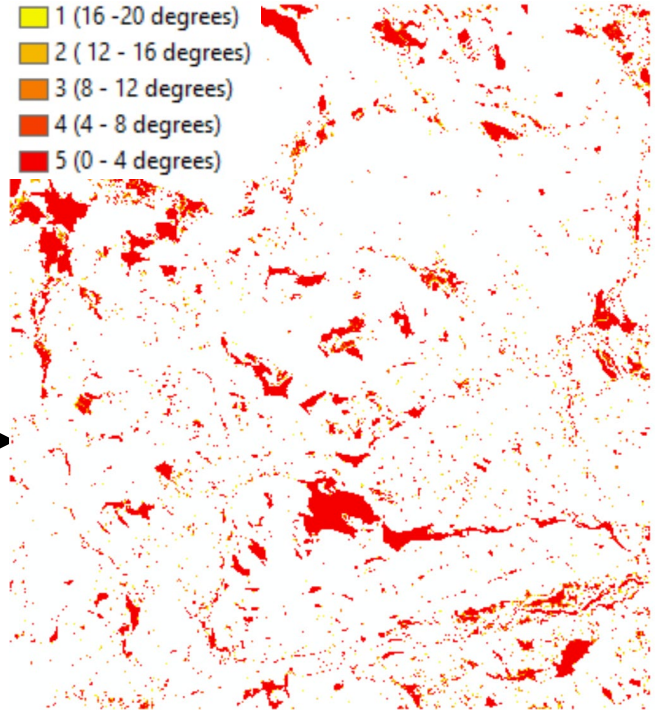


Figure 2c. slope_range

Description: assign values based on slope of surface
Source: RECLASSIFY(*SurfaceWater*)

Criterion 3: Shadow

We know for sure that the Hiker is not within any shadow. This can be done via HILLSHADE and its subsequent model shadow function. We use NOAA's Solar Position Calculator to find the solar azimuth. The following assumptions are made: location – Maine, time – 10:30 AM (late morning), date – January 1 (winter), which generated an azimuth value of 161.71. After calculating the shadow range, we can use RECLASSIFY to keep only values that have a Hillshade Value above 220 (high brightness).

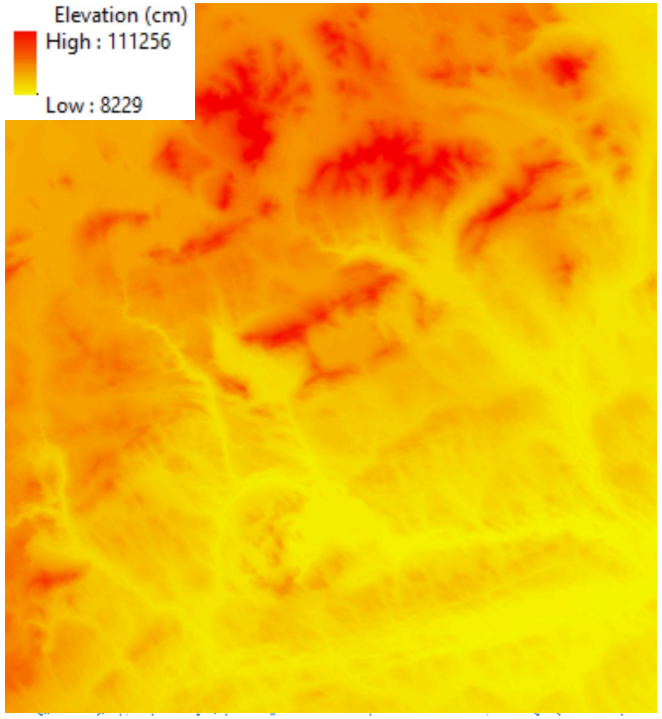


Figure 3a. Elevation
Description: elevation
Source: provided in sample

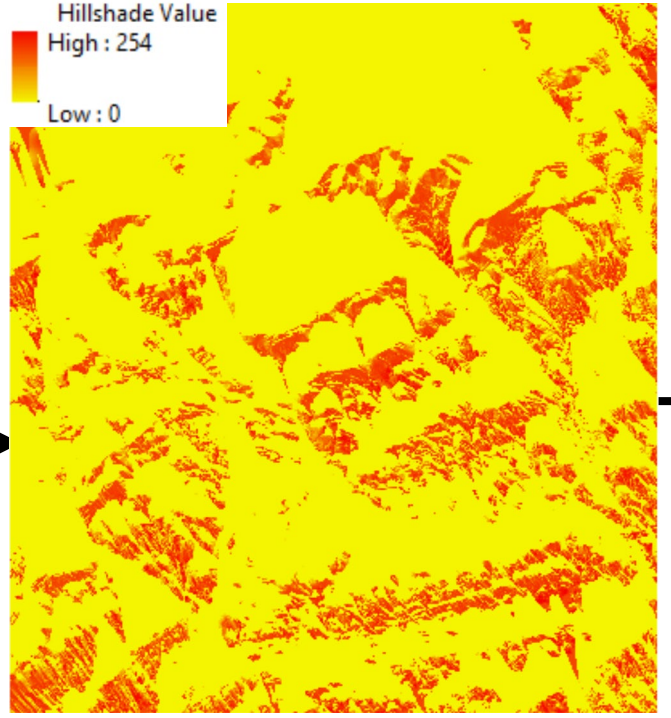


Figure 3b. shadow
Description: hillshade values of elevation
Source: HILLSHADE(*Elevation*)
Azimuth = 161.71

Figure 3c. shadow_tf
Description: hillshade values greater than 220
Source: RECLASSIFY(*Eshadow*)
0 – 220 → noData
220 - 254→ 1

Criterion 4: Near Forest Boundaries

We can also see from the picture that the hiker is near a forest. We will now need to find areas that are near both forests and mountains. First, let us consider only forest. We can obtain a forest true-false cover by using RECLASSIFY. Then, we can find distance to forests by EUCLIDIAN DISTANCE

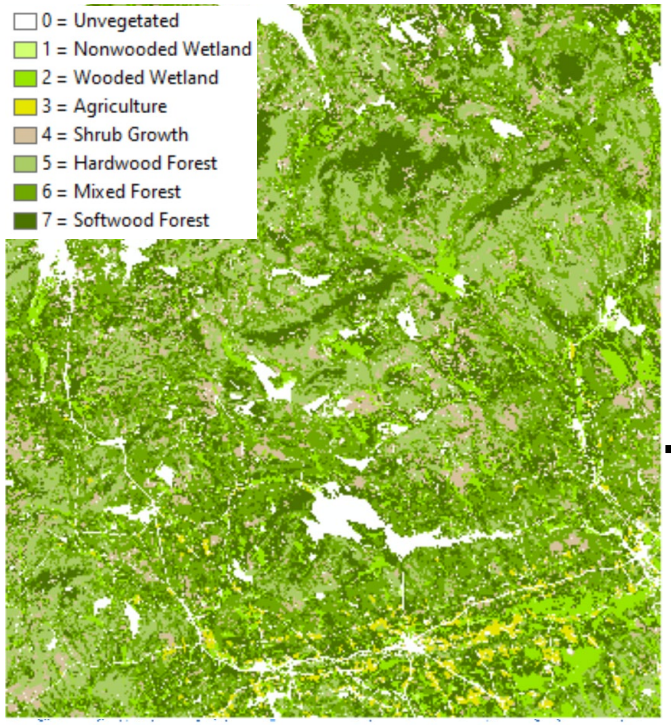


Figure 4a. Vegetation
Description: vegetation
Source: provided in sample

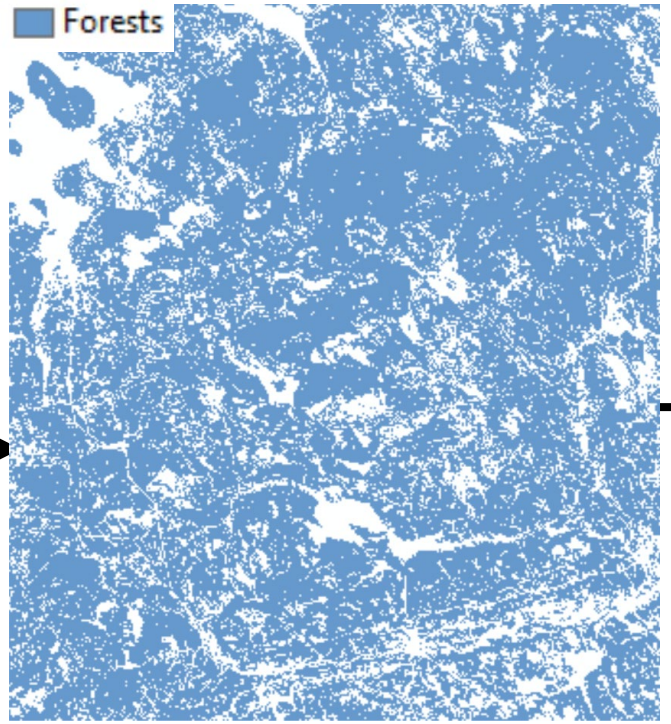


Figure 4b. forest
Description: forest true false map
Source: RECLASSIFY(*Vegetation)*
Unvegetated, Nonwood Wetland, Wooded Wetland, Agriculture, Shrub → noData
Hardwood, Mixed, Softwood Forest → 1

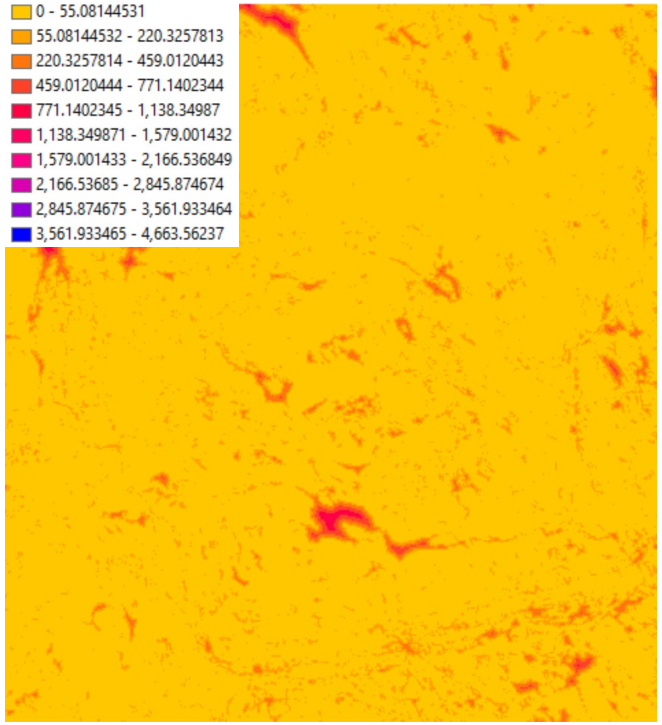


Figure 4c. forest_dist
Description: distance from forest
Source: EUCLIDIAN DISTANCE(*forest*)

Criterion 5: Near Mountain Boundaries

The hiker is near a mountain. First, let's find the FOCAL STATISTICS – MEDIAN on the elevation. Next, use RASTER CALCULATOR to do Elevation – Median to find a pixel's difference from its surroundings. Points with a higher difference are most likely peaks.

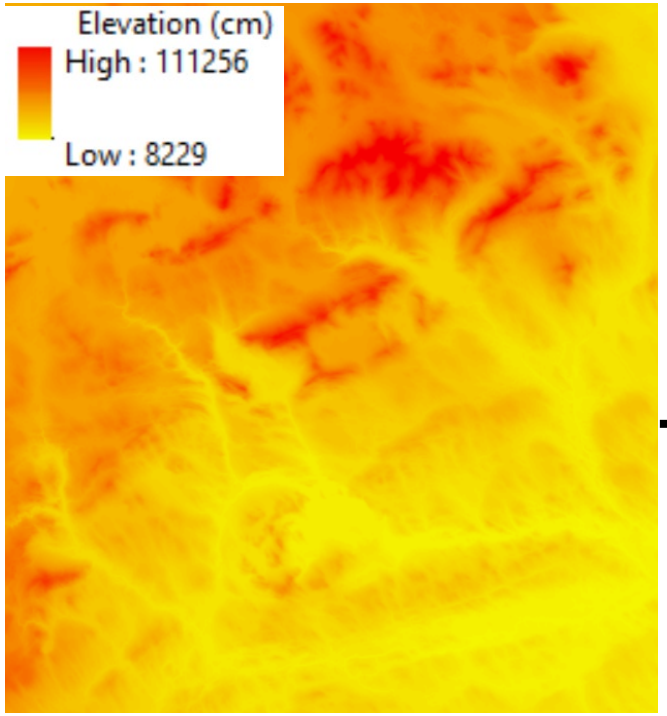


Figure 5a. Elevation
Description: elevation
Source: provided in sample

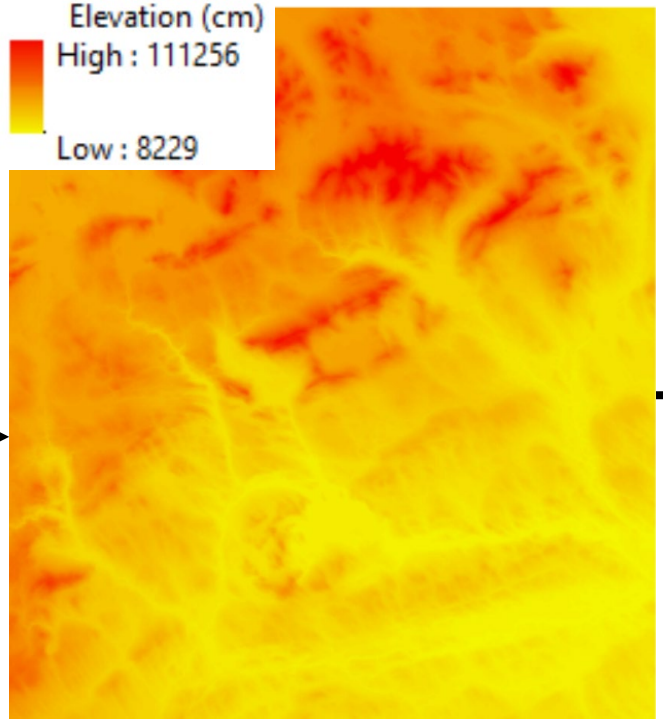


Figure 5b. elev_median
Description: median elevation from surrounding
Source: FOCAL STATISTICS – MEDIAN (*elevation*)

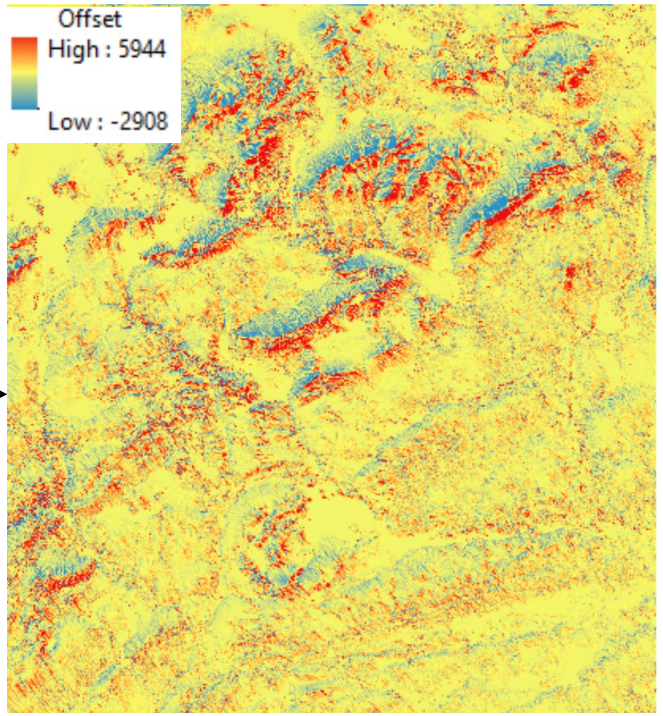


Figure 5c. elev_offset
Description: median elevation from surrounding
Source: FOCAL STATISTICS – MEDIAN (*elevation*)

Criterion 5: Near Mountain Boundaries

Figure 5d. mountains

Description: To find, mountains, let's only keep values from *elevation* that are much higher than its surroundings (or, have high *elev_offset* values)
Source: RECLASSIFY(*elev_offset*)

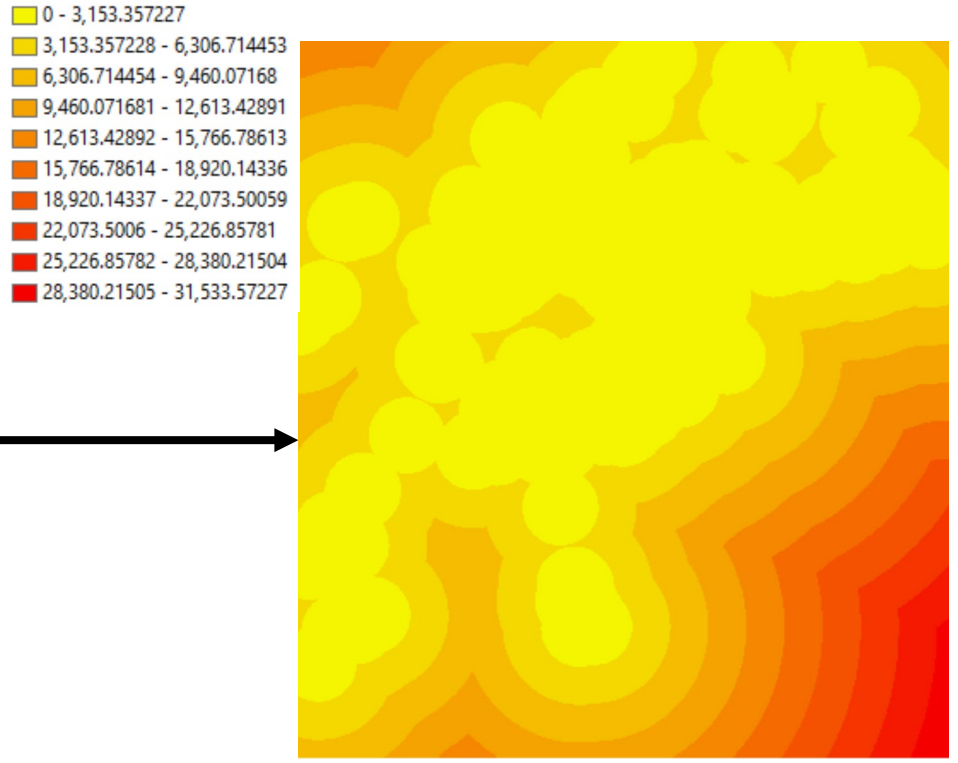


Figure 5c. mount_dist

Description: distance from mountains
Source: EUCLIDIAN DISTANCE(*mountains*)

Criterion 4/5: Forests & Mountain Distances

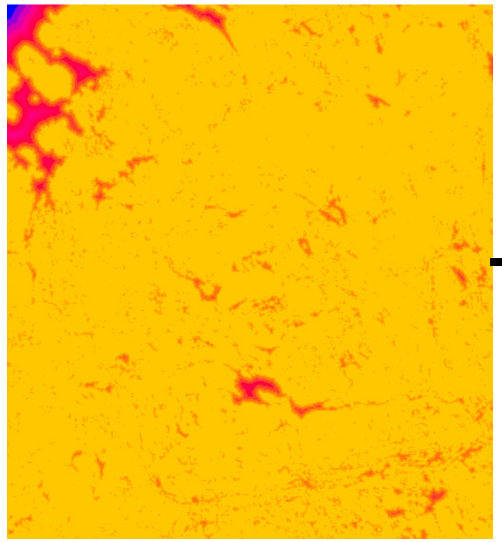


Figure 4c. forest_dist
Description: distance from forest
Source: EUCLIDIAN DISTANCE(*forest*)

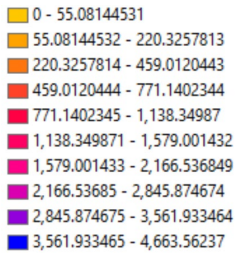
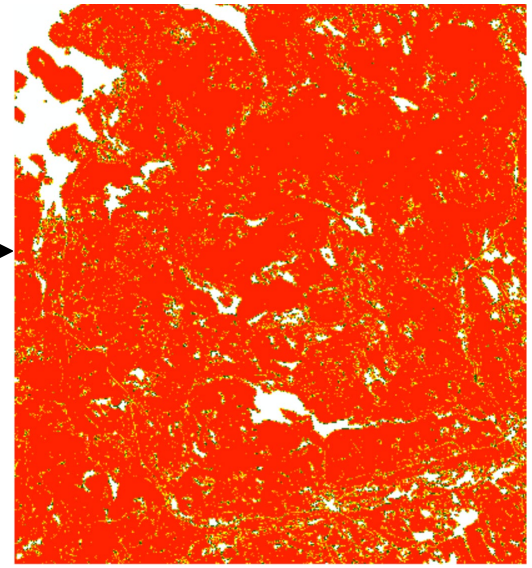
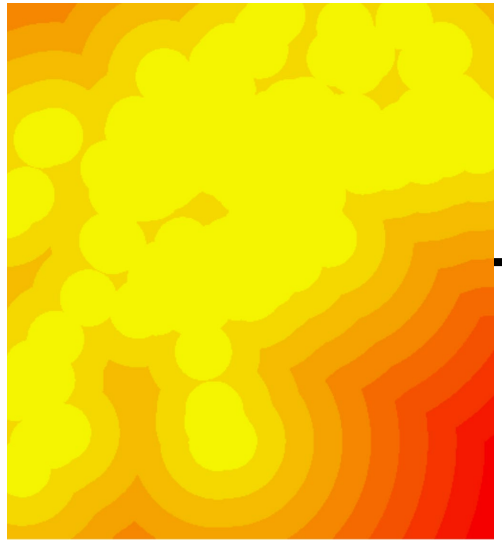


Figure 4h. for_dist_rn
Description: reclassified distance from forests
Source: RECLASSIFY(*forest_dist*)
0 – 25 → 5; 25 – 50 → 4; 50 – 75→ 3; 75 – 100 → 2; 100 – 125 → 1; >125 → noData





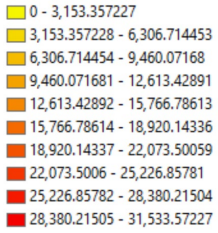


Figure 5h. mount_dist_rn
Description: reclassified distance from mountains
Source: RECLASSIFY(*mount_dist*)
0 – 2500 → 5; 2500 – 5000 → 4; 5000 – 7500→ 3; 7500 – 10000 → 2; 10000 – 12500 → 1; >12500 → noData

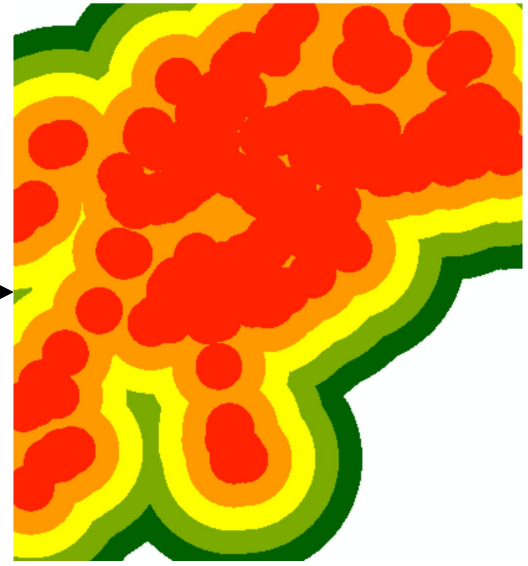
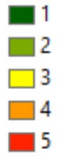


Figure 5c. mount_dist
Description: distance from mountains
Source: EUCLIDIAN DISTANCE(*mountains*)



Create classified range map of distances. Forest distances are smaller as hiker is closer to forest.

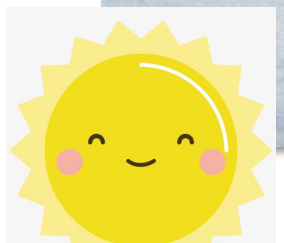
Criterion 6: Forest and Mountain Direction

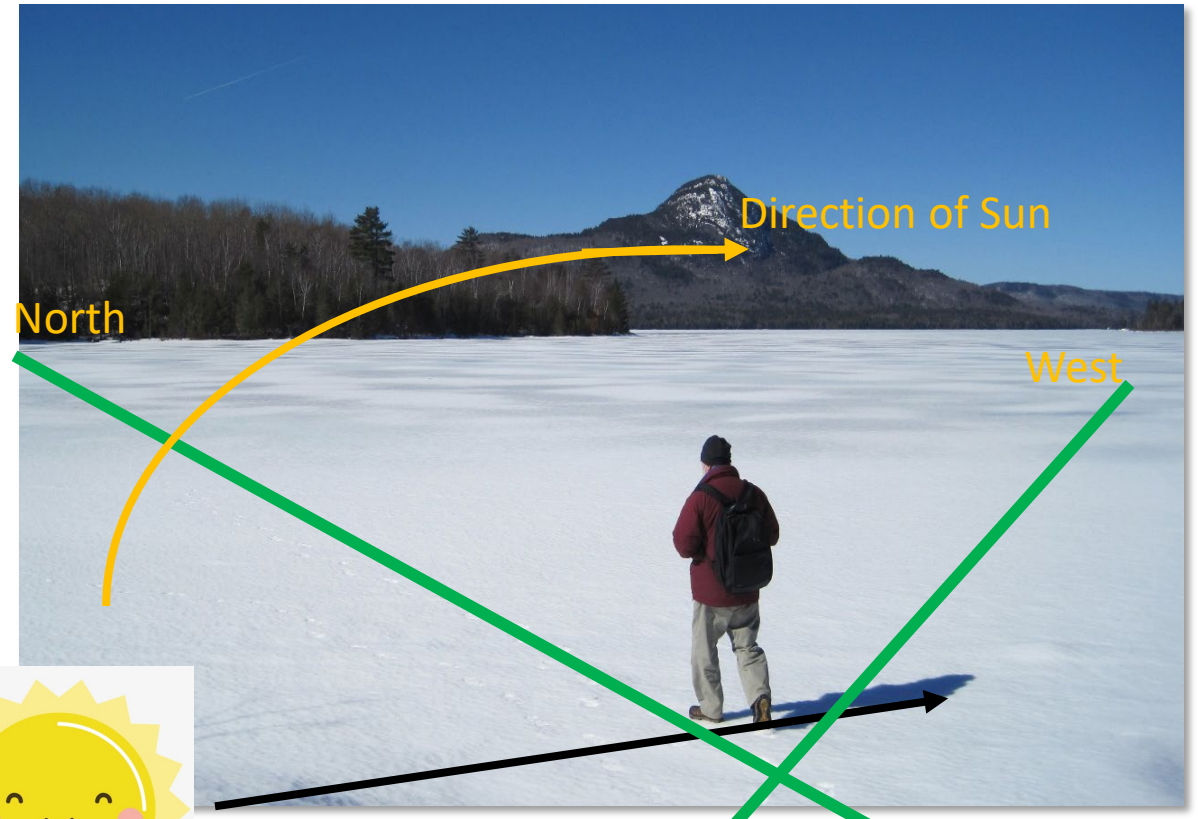
Based on the direction of the shadow and time, we can attempt to approximate the direction this hiker is facing. We will assume it is January (based on snow) and 10:30 (late morning).

We can first place the sun (see image) by placing it parallel to the shadow.

Since it is in the morning, we know that the sun is still on the eastern side of straight overhead, and is flowing from east to west. In addition, since it is almost noon, we know the sun must be positioned between the east and north, and close to north.

Thus, we can create a compass-like orientation for where the man is relative to the mountain. The Forest and Mountain are NorthWest of the hiker.





Criterion 6: Forest and Mountain Direction

We'll assume that the forest and mountain seen are the closest forest and mountain nearby. We can use EUCLIDIAN DIRECTION to find where each pixel is located relative to lakes and mountains. We also know the hiker is Southeast of both the forest and mountain. We will use RECLASSIFY to create true and false maps for pixel's direction from forests and mountains.

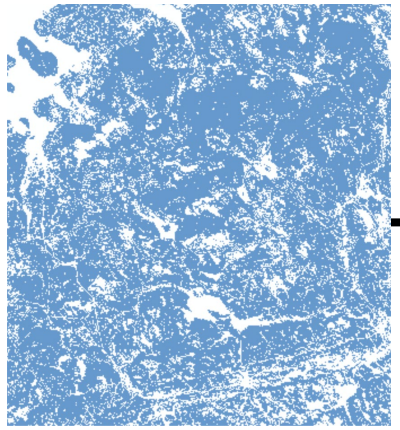


Figure 4b. forest
Description: forest true false map
Source: RECLASSIFY(*Vegetation)*
Unvegetated, Nonwood Wetland, Wooded Wetland, Agriculture, Shrub → noData
Hardwood, Mixed, Softwood Forest → 1

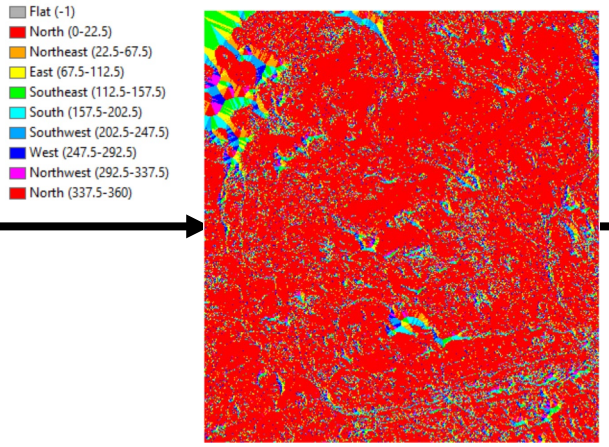


Figure 4f. forest_dir
Description: direction to nearest forest
Source: EUCLIDIAN DIRECTION *(forest_tf)*

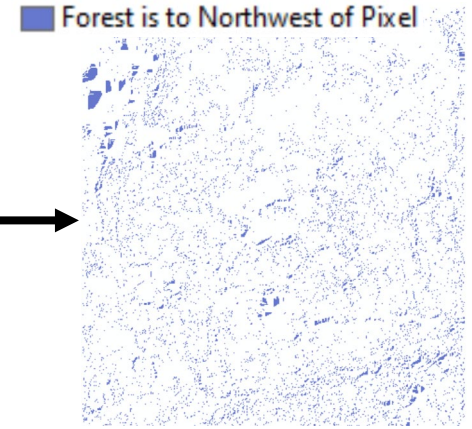


Figure 4g. forest_tf
Description: true/false for direction is south-west (270 - 360) rel to forest
Source: RECLASSIFY (*mountain_dir*)

Figure 5d. mountains
Description: To find, mountains, let's only keep values from *elevation* that are much higher than its surroundings (or, have high *elev_offset* values)
Source: RECLASSIFY(*elev_offset*)

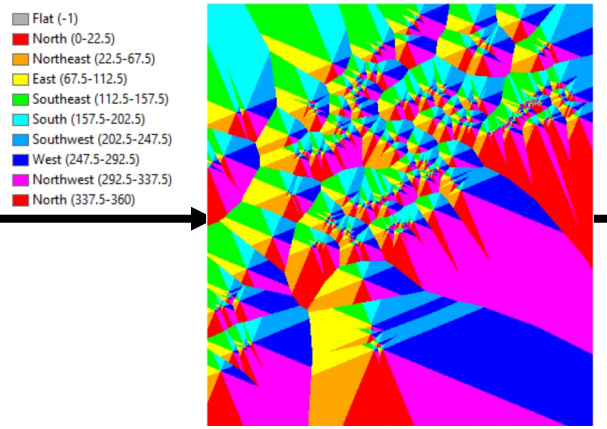


Figure 5f. moutain_dir
Description: direction to nearest mountain
Source: EUCLIDIAN DIRECTION *(mountain)*

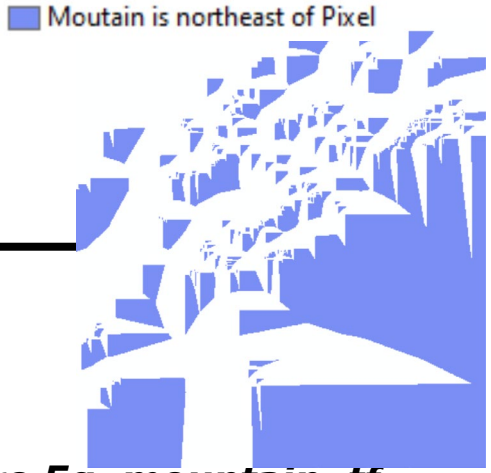


Figure 5g. mountain_tf
Description: true/false for direction is south-west (270 - 360) rel to mountain
Source: RECLASSIFY (*mountain_dir*)

There are two types of criteria to consider: The first criteria is:

- Criteria that must be fulfilled. These are true / false maps that have been produced. Any sites to be considered must be "true" for all factors
  - On Water
  - Shadow (low shadow, high brightness)
  - Forest and Mountain Direction (pixel in SOU

First, use RASTER_CALCUALTOR and run Conditional to identify pixels satisfying all requirements.

Con(("water_tf"==1) & ("shadow_tf"==1) & ("mount_tf"==1) & ("forest_tf"==1),"water_tf")

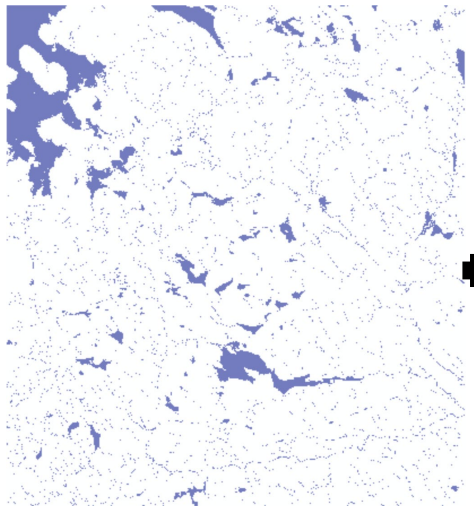


Figure 1b. water_tf
Description: true or false of water surfaces
Source: RECLASSIFY(*SurfaceWater*)

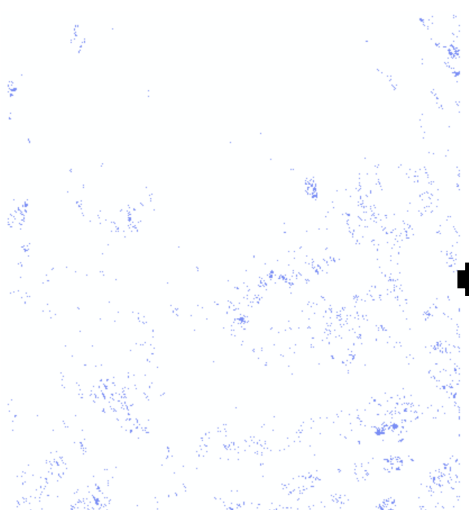


Figure 3c. shadow_tf
Description: hillshade values greater than 220
Source: RECLASSIFY(*shadow*)

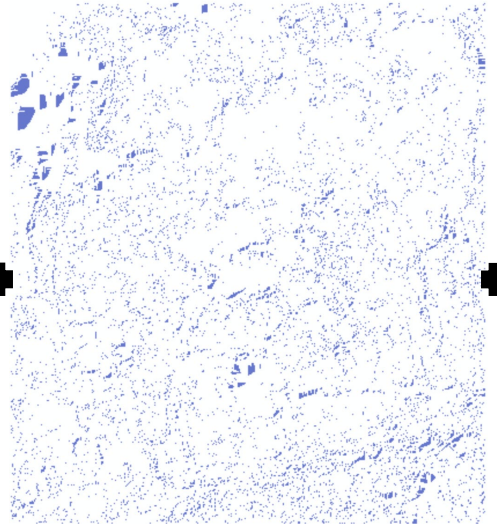


Figure 4g. mountain_tf
Description: true/false for direction is south-west (157 – 293) rel to forest

Figure 5g. mountain_tf
Description: true/false for direction is south-west (157 – 293) rel to mountain

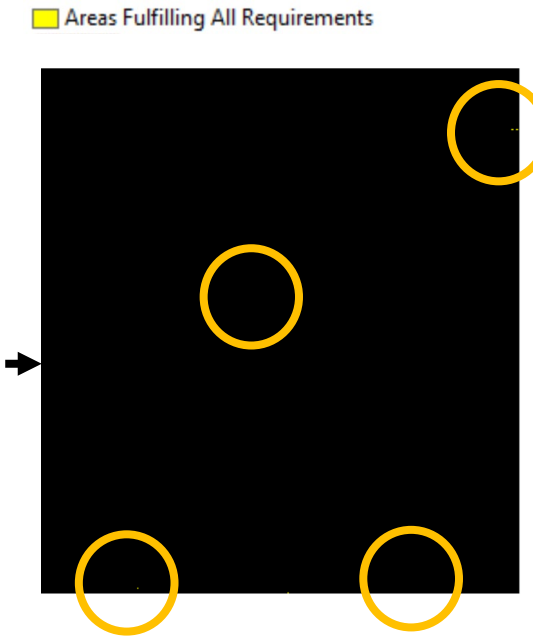


Figure 7a. total_tf
Description: true/false map for pixels fulfilling ALL requirements

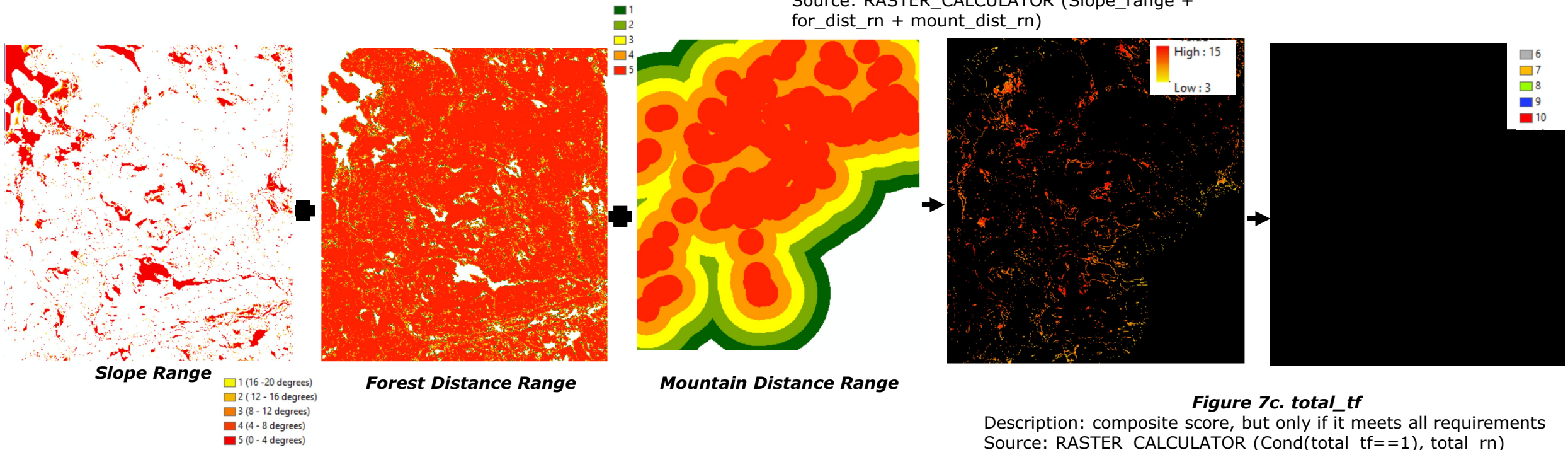
The second type of criteria is

- Criteria that are over a certain range. These are range maps, sites do not need to be considered "true" for all factors. Rather, we will extract the sites that hold the highest composite values.
  - Slope
  - Distance to Forest
  - Distance to Mountain

Use RASTER_CALCUALTOR sum the component values up

Then use RASTER_CALCULATOR Cond to extract values that fit all the true/false requirements

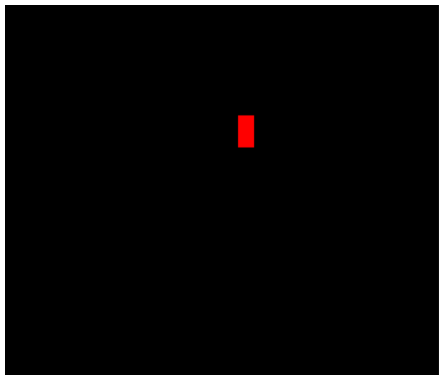
Figure 7b. total_rn
Description: composite score for three range criteria
Source: RASTER_CALCULATOR (Slope_range + for_dist_rn + mount_dist_rn)



Slope Range ***Forest Distance Range*** ***Mountain Distance Range***

Figure 7c. total_tf
Description: composite score, but only if it meets all requirements
Source: RASTER_CALCULATOR (Cond(total_tf==1), total_rn)

The max composite score is size 10. Though there doesn't seem like there are any points, on closer inspection, we can see that actually, two pixels of combined score of 10 (that much all four requirements) do exist! This is likely where the hiker is. Checking the Properties, we see there are only two pixels with values 10. If additional pixels had values of 10, there would be multiple locations the hikers could be. However, such is not the case, and we can simply use RECLASSIFY to set all values except 10 to noData



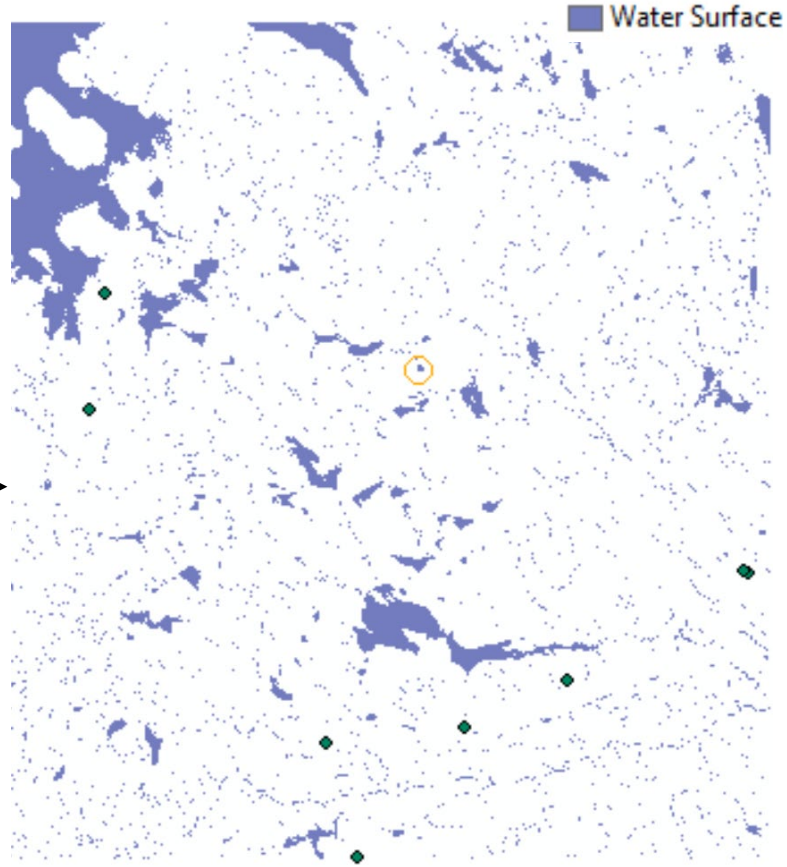
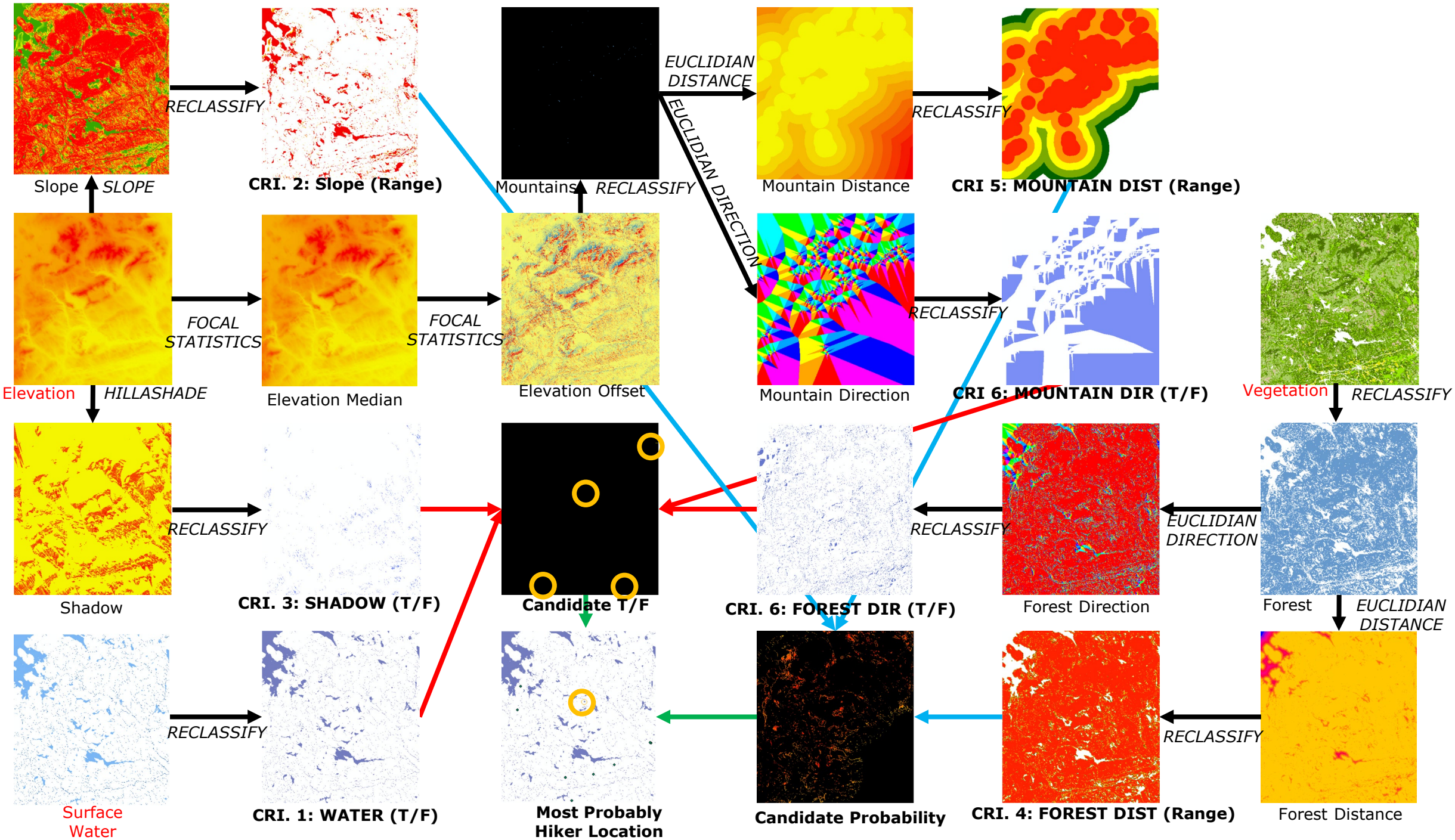


Figure 7c. total_find

Description: composite score, but only if it meets all requirements
Source: RASTER_CALCULATOR (Cond(total_tf==1), total_rn)

Figure 7d. total_max

Description: composite score for only highest value (10), null for all other regions
Source: RECLASSIFY(*total_tf*)


Slope
SLOPE
RECLASSIFY
CRI. 2: Slope (Range)
Mountains
RECLASSIFY
EUCLIDIAN DISTANCE
EUCLIDIAN DIRECTION
Mountain Distance
RECLASSIFY
CRI 5: MOUNTAIN DIST (Range)
Elevation
HILLASHADE
FOCAL STATISTICS
Elevation Median
FOCAL STATISTICS
Elevation Offset
Mountain Direction
RECLASSIFY
CRI 6: MOUNTAIN DIR (T/F)
Vegetation
RECLASSIFY
Shadow
RECLASSIFY
CRI. 3: SHADOW (T/F)
Candidate T/F
CRI. 6: FOREST DIR (T/F)
RECLASSIFY
Forest Direction
EUCLIDIAN DIRECTION
Forest
EUCLIDIAN DISTANCE
Surface Water
RECLASSIFY
CRI. 1: WATER (T/F)
Most Probably Hiker Location
Candidate Probability
CRI. 4: FOREST DIST (Range)
RECLASSIFY
Forest Distance

Option 1: More Cell Towers

Now that we have found the person, we can attempt to find where they should head to get better service. The hiker has two options here, either head to a location where he can reach more cell towers, or head closer to a cell tower. We will consider the option 1 first. Since the information in towers are in meter, we will also first convert the elevation map to meters. After converting elevation into meters, we can use VIEWSHED to visualize the reach of each cell tower.

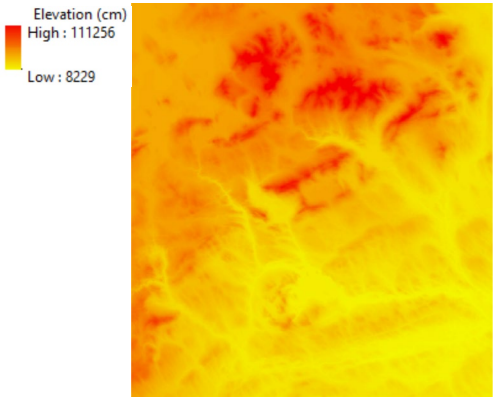


Figure 2a. elevation
Description: elevation
Source: provided in sample

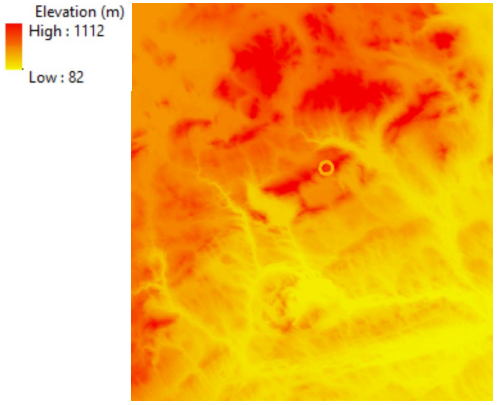


Figure 8a. elevation_m
Description: elevation to meter (floor if decimal)
Source: RASTER CALCULATOR(*elevation*/100)

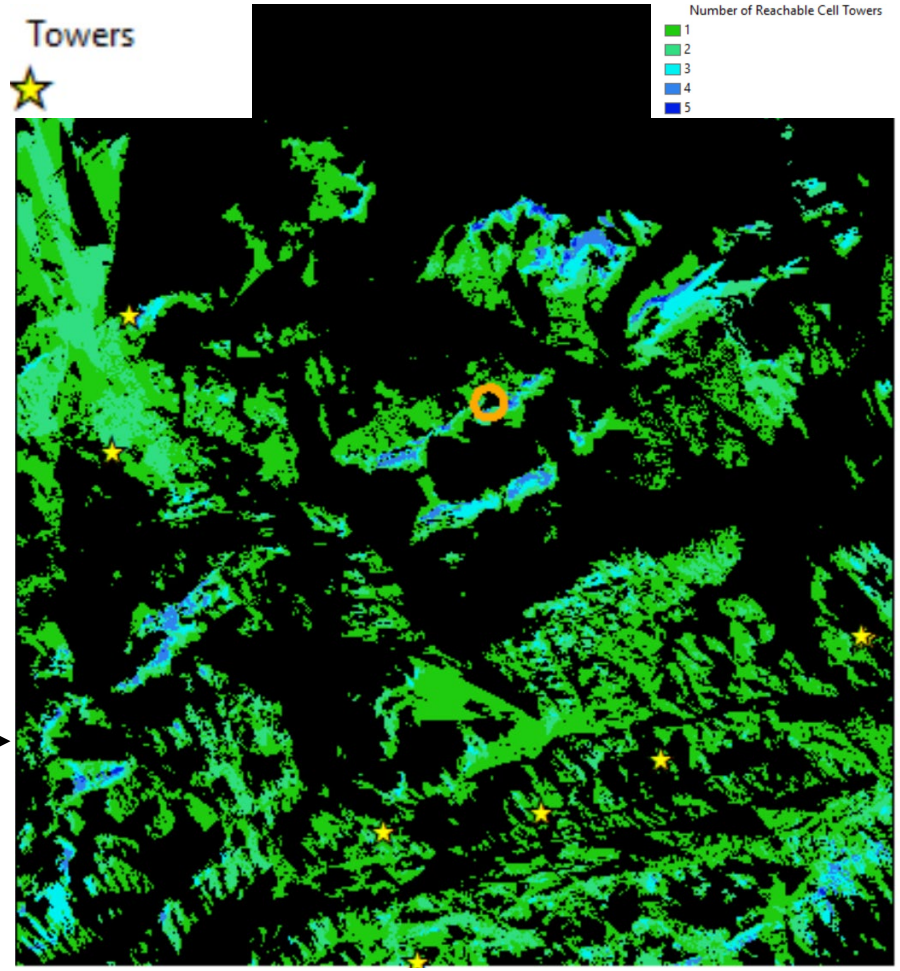
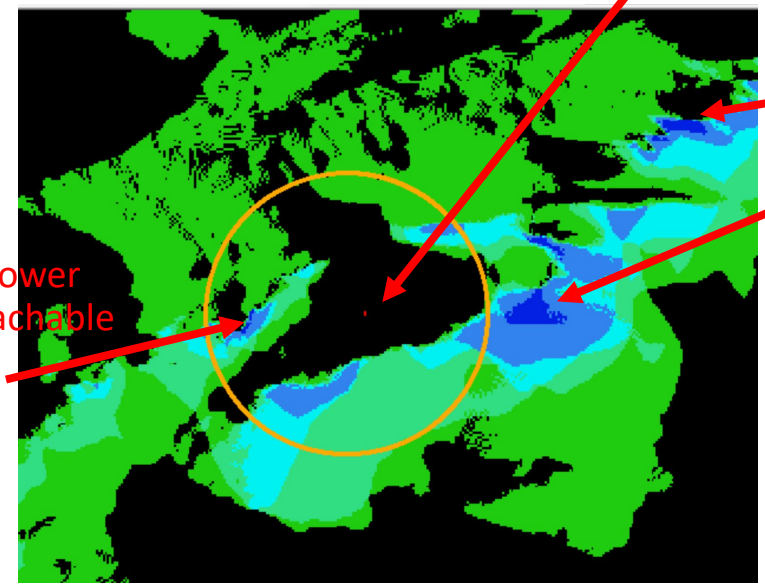


Figure 8b. Viewshed
Description: Viewshed
Source: VIEWSHED2(*towers*, *elevation*_m)



On closer inspection, we see that our hiker is actually not that far from locations reachable by 5 towers! We simply need to know find which of these regions are the closest to the hiker

Option 1: More Cell Towers

We need to find the closest 5-tower reachable region to the hiker

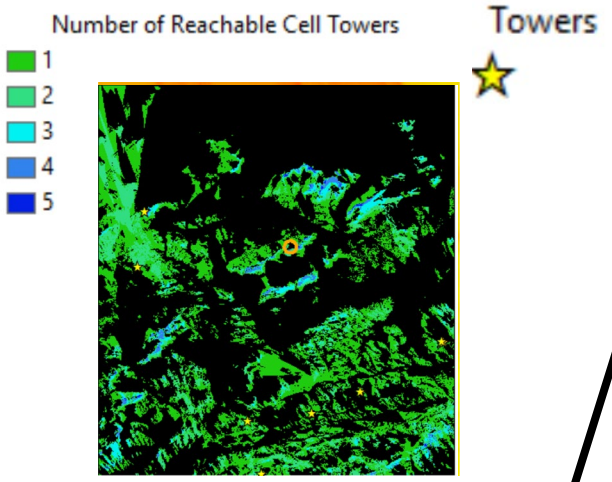


Figure 8b. viewshed
Description: Viewshed
Source: VIEWSHED2(*towers*, *elevation*_m)

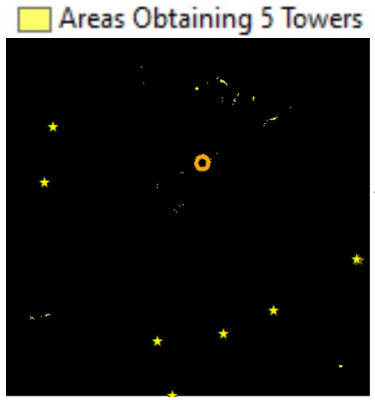


Figure 8c. five_towers
Description: areas reachable by five towers
Source: RECLASSIFY(*viewshed*)
1- 4→ noData; 5 → 1

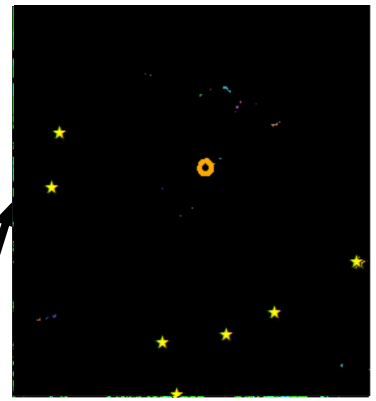


Figure 8d. 5_tower_rg
Description: region groups of 5-tower reachable regions
Source: REGION GROUP(*five_towers*)

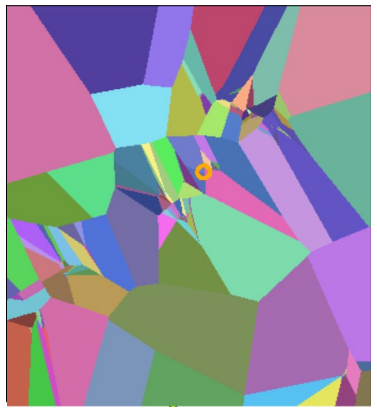
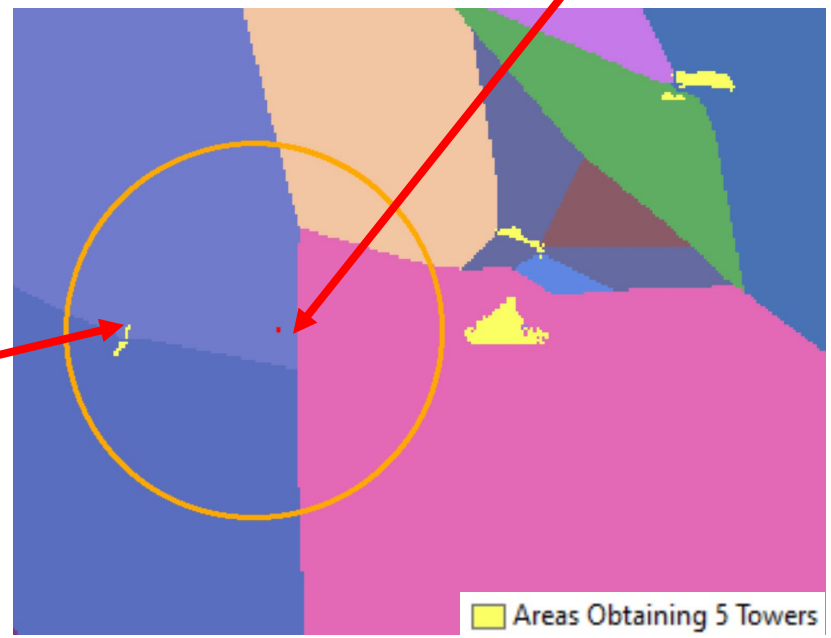


Figure 8e. 5_tower_alloc
Description: allocation to closest 5-tower region group
Source: EUCLIDIAN ALLOCATION(*5_tower_rg*)



Zoom in, and we have found where the hiker should go to get to the site that is reachable by all 5 towers!

Option 2: Closest Cell Tower

Another Option is for the hiker to get to the closest cell tower, since at the cell tower, the hiker will at the strongest reception.

We can also verify this by seeing how reception should be at each location. We can do this by using an inverse square law. First, take the EUCLIDIAN DISTANCE from tower. Then, use RASTER CALCULATOR.

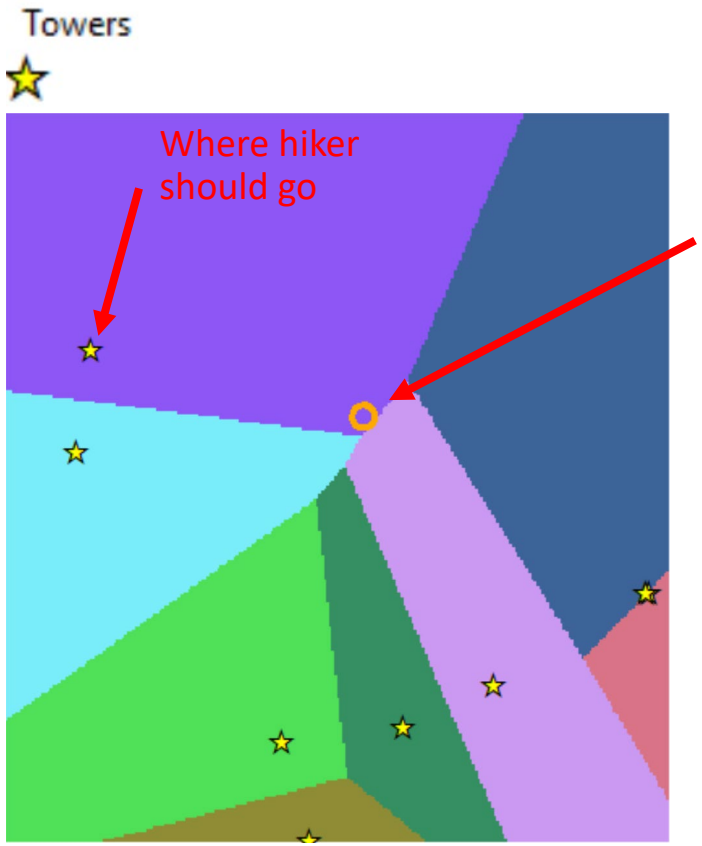


Figure 9a. Tower Alloc

Description: allocation of closest tower
Source: EUCLIDIAN ALLOCATION(*tower*)

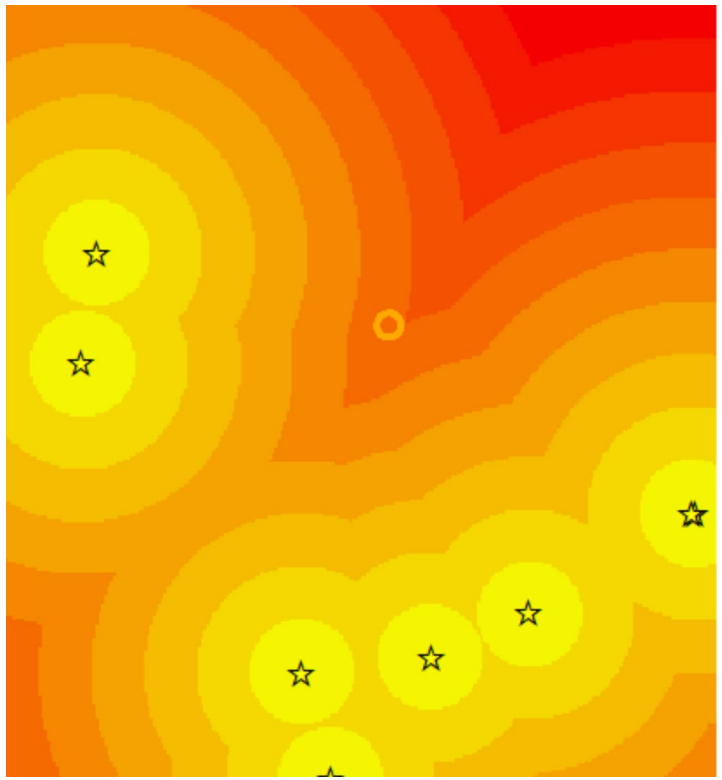


Figure 9b. tower_dist

Description: distance from nearest tower
Source: EUCLIDIAN DISTANCE(*tower*)

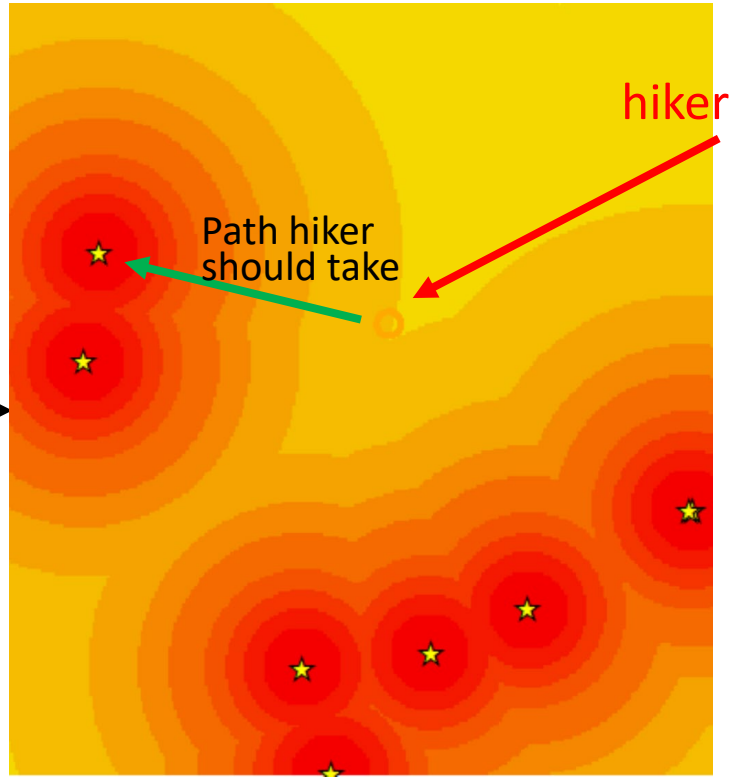


Figure 9c. signal

Description: signal from nearest tower
Source: RASTER CALCULATOR(*1*1.00/(tower_dist)*)

Conclusion

Regardless if the hiker choses option 1 or 2, he should head west.

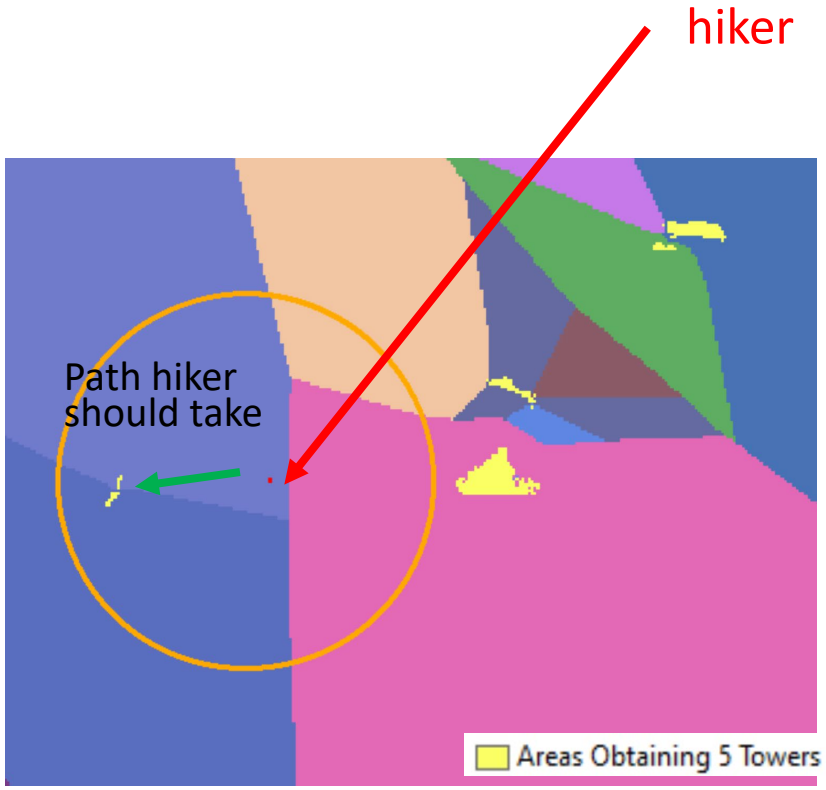


Figure 8e. 5_tower_alloc

Description: allocation to closest 5-tower region group

Source: EUCLIDIAN ALLOCATION(*5_tower_rg*)

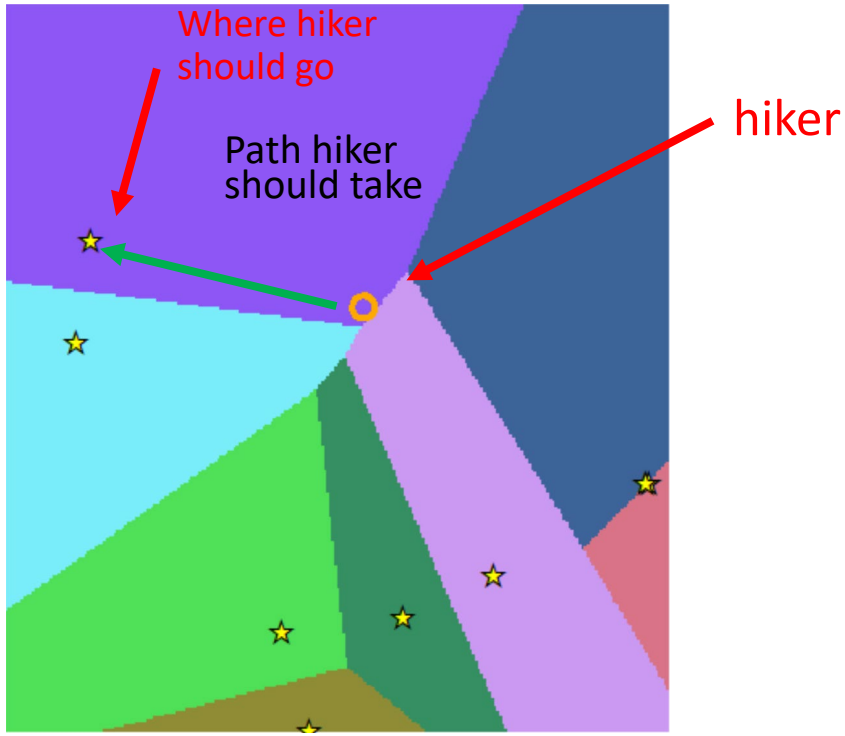


Figure 9a. Tower Alloc

Description: allocation of closest tower

Source: EUCLIDIAN ALLOCATION(*tower*)

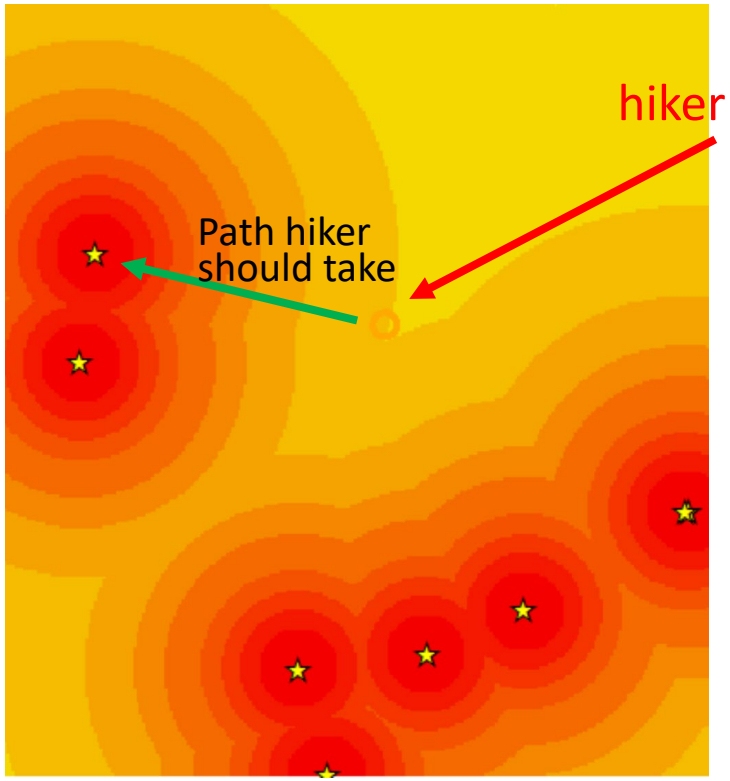


Figure 9c. signal

Description: signal from nearest tower

Source: RASTER CALCULATOR(*1*1.00/(tower_dist)*)