Homework 4 : Landscape Fragmentation

Alexandre Boucher

January 28, 2010

Due date: February 4th 2010

Compute and display key landscape metrics at the patch and class-level

Extracting patches

The patches are extracted with the (provided) function `LandscapeFragmentation`

```
def SpatialFragmentation(raster) :
```

where `raster` is a 2D array of `bool`. True indicates presence of the class type. The function returns a 2D array of integers with the shape of `raster` (each integer value is a patch ID) and a list of `Patch` type objects. The class `Patch` models a single patch.

Patch-Level Statistics

The patch level statistics are extracted with functors (a functor is an object that behave like a function, i.e. it is used by calling `()`) with a list of patches as input. In addition of the `__call__` operator the p-level class must have a `name` and `description` functions that return a string. Each functor returns a scipy array , i.e. one p-level metrics for each `Patch` object.

Class-Level Statistics

The class level statistics aggregate the information from the p-level statistics.

Deliverable

Visualization

The function `getFragStatistics` to compute, write to a file and plot user-specifed landscape metrics.

```
def getFragStatistics(patchList, plevelList, filename,
                      showPlot = True)
```

- `patchList`: List of patches (output from `LandscapeFragmentation`)
- `plevelList`: List of functors for patch processing
- `filename`: Path to the output file for the class level statistics (mean, median, standard deviation, min, max, range)
- `showPlot`: Display the histogram (and more, be creative) for each of the computed p-level metrics

Example for the class # 1 from data in `raster` 2D scipy array

```
[patchesRaster, patchesList] = SpatialFragmentation(raster == 1)
plevelStats = [PatchSizedistribution(),PatchYLength(),PatchXLength()]
getFragStatistics(patchesList, plevelStats, "myFragStat.dat", True)
```

The c-level metrics will be written in the file `myFragStat.dat` and the p-level metrics plots will be shown.

Extra Metrics

Write one more plevel metrics on your choice; call it `myPLevelMetric`. Provide a brief interpretation in the `description` field and a relevant name for the `name` field.

Data

You can test your functions on a landscape obtained through k-means. The class # 0 is only the background and is of no interest. To load the data:

```
f = open("clustersRemoteSensingK3","rb")
landscape = scipy.fromfile(file=f, dtype=scipy.int32).reshape((500,500))
f.close()
```

Note: You may have to provide the full path to the file `clustersRemoteSensingK3`.