AMERICAN MUSEUM OF NATURAL HISTORY

CENTER FOR BIODIVERSITY
AND CONSERVATION

October, 2011

Training Guide for Creating Percent Cover Images

Introduction

This guide explains how to create percent cover images using remotely sensed imagery. The tutorial will focus on creating a percent tree cover image but this approach can be used for other land cover types (i.e., shrubs, grass, bare soil, impervious surfaces) as well.

To use these instructions on your own data you will need to have one or more high resolution images which can be used to create cover masks. These data typically have a resolution of 5m or finer. You will also need to have a moderate resolution image (i.e., Landsat TM or ETM+) which will be used for the percent cover predictions to make the final image.

We will use the R open source software package to select the training data and create the percent cover image. There is a similar image classification tutorial available that illustrates how to use R to create cover masks (step 2 below) but any image processing software that supports land cover classification can be used.

Before running the scripts it is important that R is installed on your computer and you have downloaded the necessary packages. Instructions for installing R and the necessary packages can be found on the Biodiversity Informatics website.

The steps for creating percent tree cover images are listed below. These steps will be described in detail in the following sections.

1. Preprocess all of the images
2. Classify all of the high resolution images
3. Edit and run the R script to create the training data set
4. Edit and run the R script to create percent cover images
5. Optionally adjust the output image values

1. Pre-process images

The same pre-processing steps that are used in other image classification applications can also be carried out when creating percent tree cover images. Instructions for doing the specific preprocessing steps are not provided in this tutorial.

It is necessary for the high resolution images to be co-registered to the moderate resolution image. This can be a little difficult due to the potentially great difference between the resolution of the two (moderate and high resolution) types of imagery but accurate georeferencing is important.

Cloud and shadow (and other pixel values not associated with land cover such as smoke) screening of the moderate resolution image is necessary to ensure the training data coincide only with good quality data. Clouds and shadows and any no-data pixels in the moderate resolution image should be set to zero (0) or some other value that can be used to differentiate no-data from valid data. No data pixels will not be assigned a percent tree cover value. You can use a number of methods to remove clouds and shadows including manual digitizing and image classification.

You can also use radiometric correction to reduce the effect of uneven image illumination due to topography and atmospheric particulates. Radiometric corrections are not critical but they will usually improve the classification result.

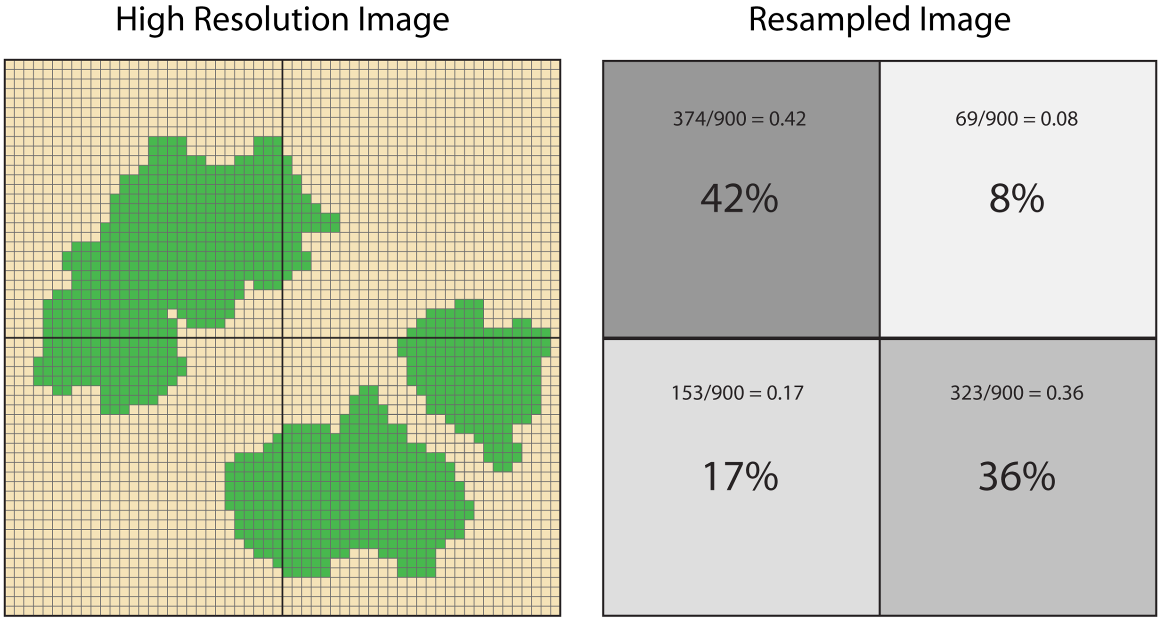
2. Classify the high resolution images

Each of the high resolution images will need to be classified into forest, non-forest, and no-data (clouds, shadows, image background) classes. You may use any image classification method that you are familiar with to create the forest, non-forest, no-data classified images.

If you are looking for information about how to create the forest/non-forest image there is a tutorial available by contacting Ned Horning (horning@amnh.org) that uses the Random Forest algorithm to create a land cover image using training polygons digitized using GIS software.

3. Edit and run the R script to create the training data set file

In this step you will create a comma separated values (CSV) file that will be used as input to the script that creates the percent cover map. This is done using a script written for R that gets the pixel values from the high resolution classified image that are co-registered to individual randomly selected moderate resolution pixels and then calculates the percent of the classified image pixels that represent your cover type of interest. In other words, if your high resolution image has a pixel size of 1m and your moderate resolution image has a pixel size of 30m the sampling process would take a block of 900 of the 1m resolution pixels that correspond to the single 30m pixel and calculate the percentage of the 1m pixels that are forest. For example, if there were 600 forest pixels and 300 non-forest pixels the value given for the output pixel would be 0.67 since 67% of the block of 1m pixels were forest. This process is illustrated in the figure below. The brown color represents non-forest and green represents forest. The high resolution image is 60 pixels x 60 pixels and is equivalent in area to four sample pixels (they are adjacent in this example but in the script the moderate resolution pixels are selected at random) with the same resolution as the moderate resolution image. Using the example above, for each image pixel a block of 900 (30 x 30) pixels are sampled from the high resolution image and the proportion of forest to non-forest is calculated. In the upper left corner of the resampled image there are 374 forest pixels and 526 non-forest pixels so the output value is 0.42 (374 forest pixels / 900 total pixels).



374/900 = 0.42 42%	69/900 = 0.08 8%
153/900 = 0.17 17%	323/900 = 0.36 36%

Subsampling a high resolution imgage (1m) to a moderate resolution image (30m). Each moderate resolution grid cell contains 900 high resolution pixels. Areas in green represent forest classified pixels. Grid on right shows the ratio of forest subsampled to the 30m image grid.

If there are clouds or other no-data values in the high resolution image the following logic will apply. If the total no-data values for a block of high resolution pixels is greater than or equal to a user defined threshold (we will use 10% i.e., 90 or more pixels in our example above) then it will not be included in the training data set since there is too much missing data to provide a reliable cover percentage. If the cloud cover is less then 10% the no-data pixels are removed from the total number of pixels when calculating the percent forest cover. For instance, using the example above if 5% of the pixels in a particular block of high resolution pixels are clouds (i.e., 45 pixels) and there are 200 forest pixels the percent forest coverage would be 23.4 (i.e., 200 / [900 – 45]).

By editing the R script "percentCoverResample.R" you are able to customize it for your own application. Near the top of the script there are a number of attributes that can be changed to customize it for your application. Here is a list of the attributes with an explanation about the attribute:

numSamps: This is the number of samples that will be selected initially. If a sample includes too many no-data values from the high-resolution classified image that sample will not be output and the resulting total number of samples in the CSV file will be less than the value specified by "numSamps".
inClassImage: Path and name for the classified image
inPredImage: Path and name for the input image that will be used for predictions
ndPred: No data value for the prediction image (inPredImage)

fromVals: The list of pixel (class) values in the classified image (inClassImage)
toVals: The values that the class values (fromVals) will be mapped to using the following rules (each value in "fromVals" must have a corresponding value in "toVals):
0 = no data such as background, clouds and shadow
1 = class for which percent cover is being calculated
2 = all other land cover classes
noDataPct: Threshold for no-data to determine if a block of high-resolution classified image pixels should be classified.
outFile: Path and name for the output CSV file

Notes: When specifying the directory path in R for a Windows computer it is necessary to use a double-backslash ("\\") instead of a single backslash ("\") or you can use a forward slash ("/") which will also work on Apple and Linux operating systems. For example the directory path: C:\Data would be typed C:\\Data or C:/Data.

R is case sensitive. In other words "a" is different from "A". Make sure the directory path and file name is exactly as it appears when you list the directory contents. Also, note that in R variable or file names should not start with a number or most special characters. It is best to start variable names with an upper or lower-case letter.

Other parts of the script can also be modified but you will need to have a good understanding of how the different commands work. The script file must be saved as an ASCII text file.

To run the script in Windows using the interface that comes with R click on the window where the script is displayed to make the window active and then from the R menu select Edit => Run all. This will execute the script and progress can be monitored in the R Console window. Another option (required when using Linux) is to launch the script from the R Console using this syntax: source("path and filename"). For example, type in the following text: source("C:\\IPY\\R_scripts\\rf_percent_cover_dual.R").

Remember to use double backslash instead of single backslash in the directory path unless you are running the script in Linux in which case you would use a single forward slash.

When the processing starts messages are printed in the R console. With several thousand sample points it can take several minutes to complete. When the script is finished you will need to check the CSV file to see if it is okay. You can view the CSV file in any text editor or spreadsheet program like Excel. There will likely be fewer sample points than what was specified in the script using the "numSamps" variable. This is because no-data values are removed. If you want more samples increase the value of "numSamps" and run the script again.

If you want to use more than one high-resolution classified image you will need to append the different output CSV files into a single CSV file. This can be done with a text editor or spreadsheet program like Excel.

4. Edit and run the R script to create percent cover images

The random forest classification process is controlled through the use of a script – “rf_percent_cover.R”. This script uses the percent cover data CSV file created in the previous step to select the predictor variables from the input image and then build a random forests regression model. The model is then used to predict percent cover over the entire input image. The output image has the percent cover values assigned to each pixel so the data range will be between 0 and 1.

By modifying this script you are able to customize it for your own application. Here is a list of the attributes with an explanation about the attribute:

pointData = Path and name for the CSV file containing the training data. This is the file that was output in the previous step.
inImage = Name and path for the input moderate resolution image. Any GDAL support format will work (i.e., GeoTiff, ERDAS .img, ENVI). A complete list is available at: http://www.gdal.org/formats_list.html.
outImage = Name and path of the output percent cover image in GeoTiff format. If you want something other than a GeoTIFF format for the output image you will need to change the ”filetype='GTiff'” statement toward the end of the script to one of the variable name for another GDAL supported format.
nd = No data value for the satellite image

The notes mentioned in the previous step are important here as well. Run the scrip using the same procedure mentioned in the previous step.

When the processing starts messages are printed in the R console and once the image classification step begins a status bar is displayed so you can monitor the progress. Large images can take several hours to process. When the classification is finished you will need to check the result to see if it is okay. You can view the result in any viewer that reads GeoTIFF (or another format if you modified the “filetype” parameter.

5. Optionally adjust the output image values

When used for regression the random forests algorithm will overestimate low values and underestimate high values. An R script (“CorrectRF_Result.R”) has been written to adjust this effect by calculating regression coefficients for the actual percent cover values from the percent cover data CSV file (produced in step 3 above) and the corresponding predicted percent cover value from the image output in the previous step. A new percent cover image is output after applying gain and offset (slope and intercept) values to the original predicted image images.

Appendices:

A – Citations and license information

If you cite this document we ask that you include the following information:
Horning, N. 2011. Training Guide for Creating Percent Cover Images. American Museum of Natural History - v10, Center for Biodiversity and Conservation. Available from http://biodiversityinformatics.amnh.org/. (accessed on *the date*).

B – Acknowledgements

We would like the thank Jeff Silverman for creating the resampling figure and for comments on the guide.

We would also like to thank the following foundations for supporting this effort:
The Google Foundation
The John D. and Catherine T. MacArthur Foundation
The Gordon and Betty Moore Foundation
The David and Lucile Packard Foundation