Readme for image analysis using Matlab.

Most of the functions described use parallel processing. Start all code with

matlabpool('open',4);

End with

matlabpool('close');

**1)** Create a binary image (matrix) to analyse. Vegetation = 1; Bare ground = 0

a) For an aerial raw image, convert the image to grayscale before filtering out required parts.

image = rgb2gray(image);

imhist(image); % filtering threshold can be identified

% as and when required, certain peaks in the histogram can be displaced. For example, if we have trees and bushes with trees close to intensity 0 and bushes greater than 50, we can equate all trees to bare ground with

image(image < 50) = 255;

image = im2bw(image, threshold = 0.5); % as colours go from 0 to 255, values less than 128 will become 0 while above 128 will become 1

image = 1 – image; % remember vegetation has to 1 and not vice versa

b) An image derived from GIS can anyway be made binary. Remember to invert colours if required.

**2)** Useful functions to clean up the image:

bwareaopen(image,30,4); % this will remove all connected components with less than 30 constituent pixels

bwmorph(image,’fill’); % this will fill all aberrant “holes” in the image (0 surrounded by 1s)

imcrop()

**3)** Smoothen the binary image for different scales of connectivity (if required)

image = connpatch(image,2);

2 is the maximum number of empty pixels between filled pixels which will still mean the filled pixels belong to the same patch. It can take any integer value.

**4)** Calculate patch size distributions

cc = bwconncomp(image,4);

psize = cellfun(@numel,cc.PixelIdxList);

psize is the vector of patch sizes. The connectivity neighbourhood here is 4 and can be changed to 8

**5)** Calculate distance to nearest neighbouring patch

intpatch = shortpatchdis(image);

intpatch is the vector of distances from each patch to the nearest neighbouring patch.

**6)** Distance between two vegetation types or trees and shrubs

Two binary images are needed in this case with the 1s representing the required vegetation type. The distances are from each object in image1.

treebush = treeshrubdis(image1,image2);

treebush is the vector of the shortest distances from each object in image1 to an object in image2

**7)** Simulate a random spread of the same patch size distributions

simimage = patchsim(image);

simimage is a new image with randomly distributed patches with the same size distribution. This new image can be analysed and compared with the original one.

**8)** Calculate the proportion of area filled within any distance from each patch

[a b] = perdfill(image,3);

Here the distance from each patch is 3 but this can take any integer value

a is the vector of proportions of vegetation while b is the vector of associated overall areas in number of pixels

**9)** For analysis, each of these distributions have to be sent to R.

csvwrite('C:/intpatch.csv',intpatch(:));

**10)** To load this data in R,

data = read.csv("C:/intpatch.csv")

data = as.vector(data[,1])

The data is ready for analysis.