

Step 2.3 Computing spectral indices

Diane ESPEL

2025-06-18

Contents

1	Objectives	1
2	Script explanation	1
2.1	Clean environment and graphics	1
2.2	Load required packages	1
2.3	Create functions	2
2.4	Define global variables	3
2.5	Set working directory	4
2.6	Load input MS raster per band	4
2.7	Run SpectralIndices() function	4

1 Objectives

Vegetation and soil moisture indices can aid in detecting terrestrial habitats. This script calculates seven spectral indices considered most relevant in the literature, using the spectral bands of a multispectral image and the developed `SpectralIndices()` function.

2 Script explanation

2.1 Clean environment and graphics

```
rm(list = ls()) # Clear all objects from the R environment to start fresh
graphics.off() # Close all graphics devices (if any plots are open)
```

2.2 Load required packages

```
library(sp) # For spatial vector data compatibility
library(terra) # For raster data processing and spatial analysis
library(stats) # For basic statistical functions (usually loaded by default)
```

2.3 Create functions

We need to declare a `SpectralIndices()` function that computes and exports a set of commonly used spectral indices from multispectral imagery, using the red, green, blue, and near-infrared (NIR) bands. Specifically, the function calculates seven indices:

- **Normalized difference vegetation index** (Pettorelli, 2013): $NDVI = \frac{(NIR-R)}{(NIR+R)}$
- **(Normalized) Green red vegetation index** (Tucker, 1979): $GRVI = \frac{(G-R)}{(G+R)}$
- **Visible atmospherically resistant index** (Gitelson et al., 2002) : $VARI = \frac{(G-R)}{(G+R+B)}$
- **Normalized Difference Water Index** (Gao, 1996): $NDWI = \frac{(G-NIR)}{(G+NIR)}$
- **Brightness index** (Khan et al., 2005): $Brightness = \sqrt{R^2 + NIR^2}$
- **Green chromatic coordinate index** (Richardson et al., 2013): $GCCI = \frac{G}{(G+R+B)}$
- **Bare soil index** (Lillo-Saavedra et al., 2018): $BSI = \frac{G+NIR}{(G-NIR)}$

Each index highlights different surface characteristics such as vegetation vigor, soil exposure, or water presence.

The function also generates a standardized filename for each output raster, and saves the resulting layers as GeoTIFF files. These indices serve as valuable predictors in habitat mapping and environmental modeling workflows.

Warning: Make sure to verify the order of spectral bands in your multispectral raster. For Pléiades/Spot6 XS imagery, the typical band order is:

- band1 = Red
- band2 = Green
- band3 = Blue
- band4 = Near-Infrared (NIR)

```
# Function to compute and export several spectral indices from MS bands
SpectralIndices <- function(archipelago, island, red, green, blue, nir, nameSat,
  year, month, resImage) {

  # Formulae for spectral indices
  ndvi_raster = (nir - red)/(nir + red) # Normalized difference vegetation index (Pettorelli,
  ↪ 2013)
  grvi_raster = (green - red)/(green + red) # Green red vegetation index (Tucker 1979)
  vari_raster = (green - red)/(green + red - blue) # Visible atmospherically resistant index
  ↪ (Gitelson et al., 2002)
  gccci_raster = green/(green + red + blue) # Green Chromatic Coordinates index (Richardson et
  ↪ al, 2013)
  brightness_raster = sqrt(red^2 + nir^2) # Brightness index (khan et al 2005)
  bsi_raster = (green + nir)/(green - nir) # Bare soil index (Lillo-Saavedra et al., 2018)
  ndwi_raster = (green - nir)/(green + nir) # Normalized Difference Water Index (Gao, 1996)

  # Recording path for spectral indices
  path_ndvi <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_", month,
    "_", resImage, "_ndvi_cut.TIF", sep = "")
```

```

path_grvi <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_", month,
  "_", resImage, "_grvi_cut.TIF", sep = "")
path_vari <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_", month,
  "_", resImage, "_vari_cut.TIF", sep = "")
path_gcci <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_", month,
  "_", resImage, "_gcci_cut.TIF", sep = "")
path_brightness <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_",
  month, "_", resImage, "_brightness_cut.TIF", sep = "")
path_bsi <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_", month,
  "_", resImage, "_bsi_cut.TIF", sep = "")
path_ndwi <- paste(archipelago, "_", island, "_", nameSat, "_", year, "_", month,
  "_", resImage, "_ndwi_cut.TIF", sep = "")

# Recording indices

writeRaster(ndvi_raster, filename = path_ndvi, overwrite = TRUE)
return(ndvi_raster)

writeRaster(grvi_raster, filename = path_grvi, overwrite = TRUE)
return(grvi_raster)

writeRaster(vari_raster, filename = path_vari, overwrite = TRUE)
return(vari_raster)

writeRaster(gcci_raster, filename = path_gcci, overwrite = TRUE)
return(gcci_raster)

writeRaster(brightness_raster, filename = path_brightness, overwrite = TRUE)
return(brightness_raster)

writeRaster(bsi_raster, filename = path_bsi, overwrite = TRUE)
return(bsi_raster)

writeRaster(ndwi_raster, filename = path_ndwi, overwrite = TRUE)
return(ndwi_raster)

}

```

2.4 Define global variables

Note: A global variable is a variable defined outside of any function. This means the variable is accessible from any part of the code, including inside functions. A global variable retains its value throughout the execution of the R script unless it is explicitly modified in the code.

It is important to define at a minimum:

- the "District": the archipelago of interest (e.g. "CRO" for Crozet archipelago)
- the "Island": the island within the archipelago of interest (e.g. "POS" for Possession island)
- the "Satellite1": the name of satellite providing multispectral imagery
- the "Year1": the acquisition year of the multispectral imagery
- the "Month1": the acquisition month of the multispectral imagery
- the "Res1": the resolution of the multispectral imagery

```

District = "CRO" # 3-letter code for archipelago (e.g. Crozet)
Island = "POS" # 3-letter code for island (e.g. Possession)

```

```
Satellite1 = "Pleiades" # satellite name of multispectral imagery
Year1 = "2022" # acquisition year of multispectral imagery
Month1 = "02" # acquisition month of multispectral imagery
Res1 = "50cm" # spatial resolution of multispectral imagery
```

2.5 Set working directory

To optimize memory, you must define one general root directory ("localscratch") that serves as the base path for your input and output data, respectively. This directory should point to the local environment where:

- input spectral bands are located under "data/raster/Cut_image"
- outputs spectral indices will be saved under "data/raster/Cut_image"

```
# Base local path (customize to your local environment) localscratch =
# paste0('your_local_path/')
localscratch = paste0("/scratch/despel/CARTOVEGE/")

# Path to open input MS raster data
open_cut_raster_path = paste0(localscratch, "data/raster/Cut_image")

# Path to save indices rasters
save_cut_raster_path = paste0(localscratch, "data/raster/Cut_image")
```

2.6 Load input MS raster per band

```
print("Loading raster bands")
R <- rast(paste0(open_cut_raster_path, "/", District, "_", Island, "_", Satellite1,
  "_", Year1, "_", Month1, "_", Res1, "_band3_cut.TIF"))
G <- rast(paste0(open_cut_raster_path, "/", District, "_", Island, "_", Satellite1,
  "_", Year1, "_", Month1, "_", Res1, "_band2_cut.TIF"))
B <- rast(paste0(open_cut_raster_path, "/", District, "_", Island, "_", Satellite1,
  "_", Year1, "_", Month1, "_", Res1, "_band1_cut.TIF"))
NIR <- rast(paste0(open_cut_raster_path, "/", District, "_", Island, "_", Satellite1,
  "_", Year1, "_", Month1, "_", Res1, "_band4_cut.TIF"))
```

2.7 Run SpectralIndices() function

```
setwd(save_cut_raster_path) # set save path working directory
print("Spectral indices are computed")
SpectralIndices(archipelago = District, island = Island, red = R, green = G, blue = B,
  nir = NIR, nameSat = Satellite1, year = Year1, month = Month1, resImage = Res1)
```