Step 2.3 Computing spectral indices

Diane ESPEL

2025-10-31

Contents

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

```r
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

```r
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)

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

  # 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)
  gcci_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
  make_path <- function(index) {
    file.path(save_path, paste0(archipelago, "_", island, "_", nameSat, "_", year, "_", month, "_",
     ↪  resImage, "_", index, "_cut.TIF"))
  }
  # Recording indices
  writeRaster(ndvi_raster, filename = make_path("ndvi"), overwrite = TRUE)
  writeRaster(grvi_raster, filename = make_path("grvi"), overwrite = TRUE)
  writeRaster(vari_raster, filename = make_path("vari"), overwrite = TRUE)
  writeRaster(gcci_raster, filename = make_path("gcci"), overwrite = TRUE)
  writeRaster(brightness_raster, filename = make_path("brightness"), overwrite = TRUE)
  writeRaster(bsi_raster, filename = make_path("bsi"), overwrite = TRUE)
  writeRaster(ndwi_raster, filename = make_path("ndwi"), overwrite = TRUE)

  # Return all created rasters
  return(list(
    NDVI = ndvi_raster,
    GRVI = grvi_raster,
    VARI = vari_raster,
    GCCI = gcci_raster,
    Brightness = brightness_raster,
    BSI = bsi_raster,
    NDWI = 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 `"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
Res1 = "50cm"  # spatial resolution of multispectral imagery
```

2.5 Set working directory

To optimize memory, you must define one general root directory (`"localHOME`) 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)
#localHOME = paste0("your_local_path/")
localHOME=paste0("/home/genouest/cnrs_umr6553/despel/CARTOVEGE/")
```

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

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

2.6 Run SpectralIndices() for each available year and month

```
# List of years and months
all_years <- c("2025","2024","2023","2022","2021","2020","2015","2011")
all_months <- c("01","02","03","04","05","06","07","08","09","10","11","12")

# Run the loop on available rasters
for (Year1 in all_years) {

  print(paste0("Year: ", Year1))

  for (Month1 in all_months) {
    print(paste0("Month: ", Month1))

    raster_path_blue <- file.path(open_cut_raster_path, paste0(District, "_", Island, "_",
 ↪  Satellite1, "_", Year1, "_", Month1, "_", Res1, "_band1_cut.tif"))
    raster_path_green <- file.path(open_cut_raster_path, paste0(District, "_", Island, "_",
 ↪  Satellite1, "_", Year1, "_", Month1, "_", Res1, "_band2_cut.tif"))
    raster_path_red <- file.path(open_cut_raster_path, paste0(District, "_", Island, "_",
 ↪  Satellite1, "_", Year1, "_", Month1, "_", Res1, "_band3_cut.tif"))
    raster_path_nir <- file.path(open_cut_raster_path, paste0(District, "_", Island, "_",
 ↪  Satellite1, "_", Year1, "_", Month1, "_", Res1, "_band4_cut.tif"))

    # Test that ALL four files exist
    all_exist <- all(file.exists(c(raster_path_blue, raster_path_green, raster_path_red,
 ↪  raster_path_nir)))

    if (!all_exist) {
      message(" One or more band files missing for ", Year1, "-", Month1, " -> skipping.")
      next
    }


    # Load rasters
    B <- rast(raster_path_blue)
    G <- rast(raster_path_green)
    R <- rast(raster_path_red)
    NIR <- rast(raster_path_nir)

    # Compute indices and write to save_path
    message(" Computing spectral indices for ", Year1, "-", Month1)

     ↪  SpectralIndices(archipelago=District,island=Island,red=R,green=G,blue=B,nir=NIR,nameSat=Satellite1,year=Yea
     ↪  = save_cut_raster_path)

  } # end of month
} # end of year
```