

# Project Part 1

Virginia Brame, Clay Harris, Hai Liu

2025-02-20

```
knitr::opts_chunk$set(echo=TRUE)
knitr::opts_chunk$set(cache=TRUE, autodep=TRUE)
knitr::opts_chunk$set(fig.align="center", fig.pos="H")

library(doParallel)
cl <- makePSOCKcluster(parallel::detectCores(logical = FALSE))
registerDoParallel(cl)

library(tidyverse)
library(tidymodels)
library(discrim)
library(leaflet)
library(terra)
library(htmlwidgets)
library(leaflet)
library(stars)
library(colordistance)
library(jpeg)
```

## Data loading

```
files <- c(
  "orthovnir057_ROI_NON_Blue_Tarps.txt",
  "orthovnir078_ROI_NON_Blue_Tarps.txt",
  "orthovnir067_ROI_NOT_Blue_Tarps.txt",
  "orthovnir069_ROI_NOT_Blue_Tarps.txt"
)

non_not_blue_data <- map_dfr(files, ~ {
  read_fwf(
    file = .x,
    col_positions = fwf_widths(
      widths = c(6, 6, 10, 12, 12, 10, 12, 5, 5, 5),
      col_names = c("ID", "X", "Y", "Map_X", "Map_Y", "Lat", "Lon", "B1", "B2", "B3")
    ),
    skip = 8
  ) %>%
  mutate(Class = str_extract(.x, "(NON_Blue_Tarps|NOT_Blue_Tarps)"))
})

blue_files <- c(
  "orthovnir069_ROI_Blue_Tarps.txt",
  "orthovnir067_ROI_Blue_Tarps.txt",
```

```

"orthovnir078_R0I_Blue_Tarps.txt"
)

blue_data <- map_dfr(blue_files, ~ {
  read_fwf(
    file = .x,
    col_positions = fwf_widths(
      widths = c(5, 6, 10, 12, 11, 10, 12, 5, 5, 5),
      col_names = c("ID", "X", "Y", "Map_X", "Map_Y", "Lat", "Lon", "B1", "B2", "B3")
    ),
    skip = 8
  ) %>%
  mutate(Class = "Blue_Tarps")
})

```

```

holdout_data <- bind_rows(
  non_not_blue_data %>% select(Lat, Lon, B1, B2, B3, Class),
  blue_data %>% select(Lat, Lon, B1, B2, B3, Class)
)

```

```
glimpse(holdout_data)
```

```

## Rows: 2,004,177
## Columns: 6
## $ Lat   <dbl> 18.52267, 18.52267, 18.52267, 18.52267, 18.52267, 18.52267, 18.5~
## $ Lon   <dbl> -72.44399, -72.44399, -72.44399, -72.44399, -72.44399, -72.44399~
## $ B1    <dbl> 104, 101, 103, 107, 109, 103, 100, 98, 97, 99, 103, 104, 101, 96~
## $ B2    <dbl> 89, 80, 87, 93, 99, 73, 79, 70, 73, 79, 84, 86, 83, 79, 77, 82, ~
## $ B3    <dbl> 63, 60, 69, 72, 68, 53, 56, 51, 56, 61, 63, 62, 61, 58, 56, 59, ~
## $ Class <chr> "NON_Blue_Tarps", "NON_Blue_Tarps", "NON_Blue_Tarps", "NON_Blue_~

```

```
# Convert for terra
```

```

holdout_data_sp <- holdout_data %>%
  rename(x = Lon, y = Lat)
v <- terra::vect(holdout_data_sp, geom = c("x", "y"), crs = "EPSG:4326")

```

```
# Reproject to UTM (resolution in meters)
```

```
v_utm <- terra::project(v, "EPSG:32618")
```

```
# Create empty raster
```

```
r_empty <- terra::rast(terra::ext(v_utm), resolution = 0.1, crs = "EPSG:32618")
```

```
# 3 bands
```

```

r_b1 <- terra::rasterize(v_utm, r_empty, field = "B1", filename = "r_b1.tif", overwrite = TRUE)
r_b2 <- terra::rasterize(v_utm, r_empty, field = "B2", filename = "r_b2.tif", overwrite = TRUE)
r_b3 <- terra::rasterize(v_utm, r_empty, field = "B3", filename = "r_b3.tif", overwrite = TRUE)

```

```
# Combine bands
```

```
rgb_raster <- c(r_b1, r_b2, r_b3)
```

```
# Reproject back to wgs84
```

```
rgb_raster_wgs <- terra::project(rgb_raster, "EPSG:4326", filename = "rgb_raster_wgs.tif", overwrite = TRUE)
```

```
# Convert to brick for leaflet
```

```
rgb_brick <- raster::brick(rgb_raster_wgs)
```

```

# Create map
#m <- leaflet() %>%
#  addTiles() %>%
#  leafem::addRasterRGB(rgb_brick, r = 1, g = 2, b = 3)

# Save the map
#htmlwidgets::saveWidget(m, "interactive_map.html")

# Aggregate raster (factor 10)
rgb_brick_coarse <- raster::aggregate(rgb_brick, fact = 10, fun = mean)

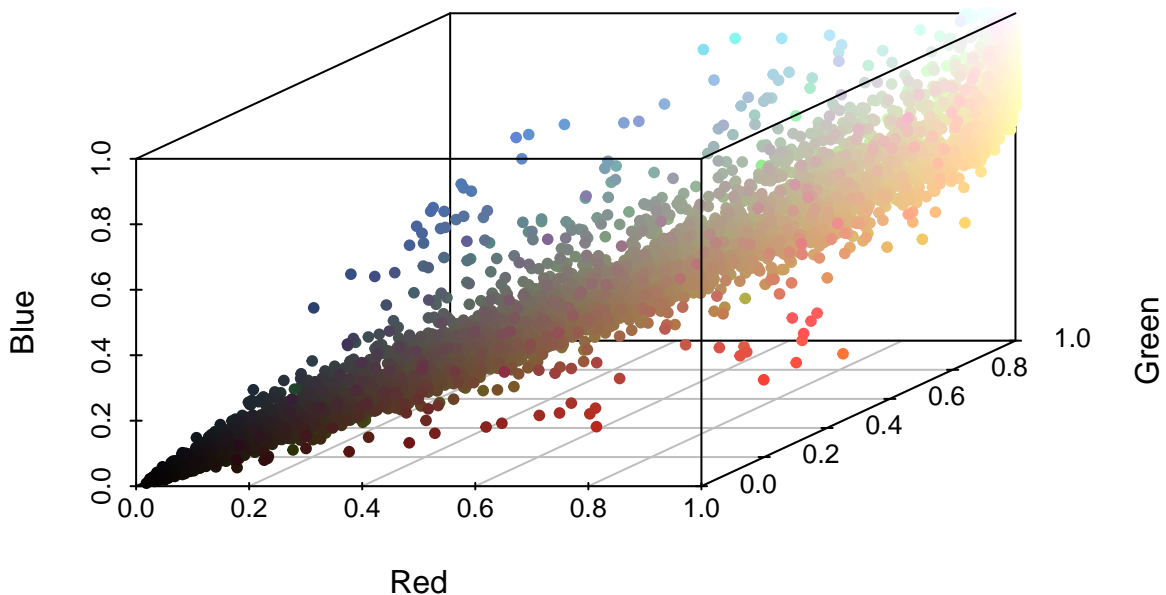
# Create map
m <- leaflet() %>%
  addTiles() %>%
  leafem::addRasterRGB(rgb_brick_coarse, r = 1, g = 2, b = 3)

htmlwidgets::saveWidget(m, "interactive_map_coarse.html")

image_path <- "orthovnir078_makeshift_villiage1.jpg"
colordistance::plotPixels(image_path)

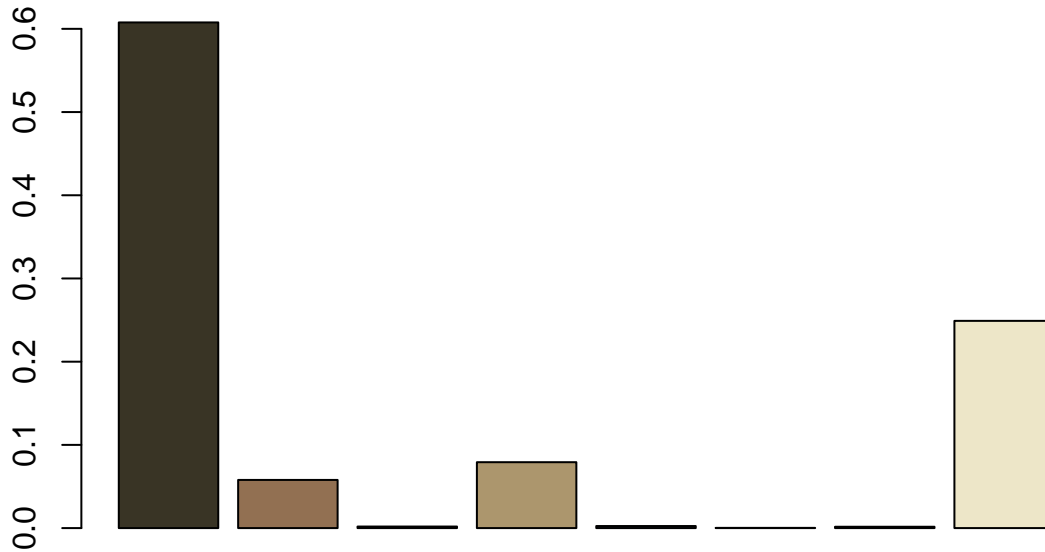
```

### orthovnir078\_makeshift\_villiage1.jpg , 10000 points



```
H8hist <- colordistance::getImageHist(image_path, bins=c(2, 2, 2))
```

## orthovnir078\_makeshift\_villiage1



```
# Number of pixels
n <- nrow(holdout_data)
maxHeight <- 65500
height <- min(n, maxHeight)
width <- ceiling(n / height)
total_pixels <- height * width

# Normalize RGB
r <- holdout_data$B1 / 255
g <- holdout_data$B2 / 255
b <- holdout_data$B3 / 255

# Calculate padding
pad <- total_pixels - n

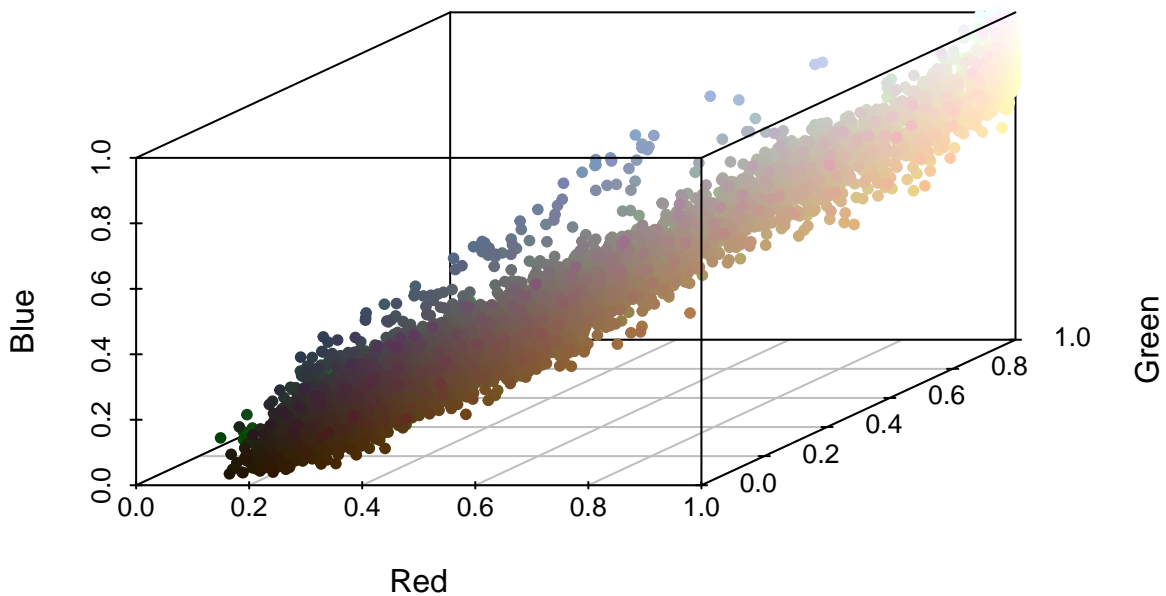
# Pad
if(pad > 0){
  r <- c(r, rep(0, pad))
  g <- c(g, rep(1, pad))
  b <- c(b, rep(0, pad))
}

# Make array
img_array <- array(c(matrix(r, nrow = height, ncol = width),
                        matrix(g, nrow = height, ncol = width),
                        matrix(b, nrow = height, ncol = width)),
                  dim = c(height, width, 3))

# Write to jpg
writeJPEG(img_array, target = "holdout_colors.jpg")

image_path <- "holdout_colors.jpg"
colordistance::plotPixels(image_path)
```

## holdout\_colors.jpg , 10000 points



```
test_data <- read.csv("HaitiPixels.csv")

# Number of pixels
n <- nrow(test_data)
maxHeight <- 65500
height <- min(n, maxHeight)
width <- ceiling(n / height)
total_pixels <- height * width

# Normalize RGB
r <- test_data$Red / 255
g <- test_data$Green / 255
b <- test_data$Blue / 255

# Calculate padding
pad <- total_pixels - n

# Pad
if(pad > 0){
  r <- c(r, rep(0, pad))
  g <- c(g, rep(1, pad))
  b <- c(b, rep(0, pad))
}

# Make array
img_array <- array(c(matrix(r, nrow = height, ncol = width),
                     matrix(g, nrow = height, ncol = width),
                     matrix(b, nrow = height, ncol = width)),
                  dim = c(height, width, 3))

# Write to jpg
writeJPEG(img_array, target = "test_colors.jpg")
```

```
image_path <- "test_colors.jpg"  
colordistance::plotPixels(image_path)
```

**test\_colors.jpg , 10000 points**

