Spectralmatch: relative radiometric normalization toolkit for raster mosaics and time series

Kanoa Lindiwe[1,2], Joseph Emile Honour Percival[1], Ryan Perroy[1]

[1]Spatial Data Analysis and Visualization Lab (SDAV) at the University of Hawai'i at Hilo
[2]Tropical Conservation Biology and Environmental Science Graduate Program at the University of Hawai'i at Hilo

This work was supported in part by the National Science Foundation under EPSCoR Research Infrastructure Improvement Award OIA-2149133 and the Hau'oli Mau Loa Foundation Graduate Scholarship.

Overview

Spectralmatch is an **open source Python library**, **command line interface**, and **QGIS Plugin** to perform relative radiometric normalization to enhance spectral consistency across raster mosaics and time series.

Two WorldView-3 images in Hawai'i: (left) before matching and (right) after processing through spectralmatch

← **Documentation | GitHub** →

Problem

Imagery is affected by inconsistencies that reduce **analytical accuracy** and the **detection of actual environmental variables**.

Ground → Atmosphere → Sensor

Topographic effects—slope, aspect, terrain shading, and elevation irradiance differences

Illumination geometries—solar zenith, solar azimuth, incidence angle, and sun–terrain orientation

Sensor view geometries—view zenith, view azimuth, off-nadir angle, and pixel-projection geometry

Surface condition anisotropy—BRDF effects, texture directionality, and moisture-driven variation

Scattering—Rayleigh, Mie, and aerosol scattering

Absorption—water vapor, ozone, and mixed gas absorption bands

Adjacency effect—radiance from nearby surfaces scattered into adjacent pixels

Path anisotropy—angle-dependent path length, terrain height, and solar-sensor geometry

Sensor calibration drift—temporal radiometric degradation, detector gain shifts

Multi-sensor spectral response mismatch—differing band centers, bandwidths, and spectral response functions

Scene-dependent sensor processing

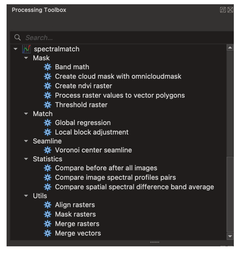
Vignetting

Usage

Python

```
global_regression(
    input_images="in",
    output_images="global",
)

local_block_adjustment(
    input_images="global",
    output_images="local",
)

align_rasters(
    input_images="local",
    output_images="aligned",
    tap=True,
)

voronoi_center_seamline(
    input_images="aligned",
    output_mask="ImageMasks.gpkg",
    image_field_name="image",
)

mask_rasters(
    input_images="aligned",
    output_images="clipped",
    vector_mask=("include", "ImageMasks.gpkg", "image"),
)

merge_rasters(
    input_images="clipped",
    output_image_path="Merged.tif",
)
```

QGIS Plugin

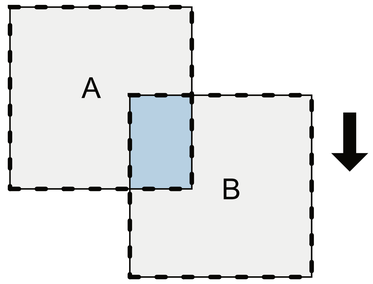


Command Line Interface

```
spectralmatch global_regression \
  input_images=in \
  output_images=global

spectralmatch local_block_adjustment \
  input_images=global \
  output_images=local

spectralmatch align_rasters \
  input_images=local \
  output_images=aligned \
  tap=True

spectralmatch voronoi_center_seamline \
  input_images=aligned \
  output_mask=ImageMasks.gpkg \
  image_field_name=image

spectralmatch mask_rasters \
  input_images=aligned \
  output_images=clipped \
  vector_mask="include,ImageMasks.gpkg,image"

spectralmatch merge_rasters \
  input_images=clipped \
  output_image_path=Merged.tif
```

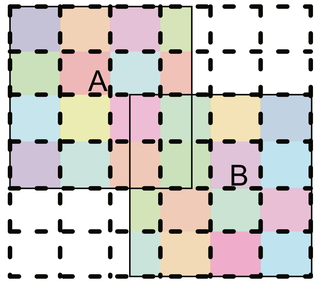
Approach

Global Regression—adjusts brightness and contrast per image



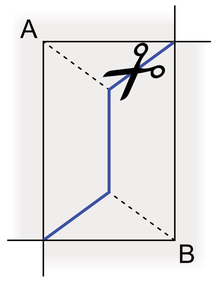
- **Extract overlapping areas**
- **Mask with Pseudo-Invariant Features**
- **Calculate mean & std per-overlap per-band**
- **Least squares to minimize the overlap difference**
- **Coefficients are used as gain & offset per-image, per-band**

Local Block Adjustment—block-wise dynamic gamma correction



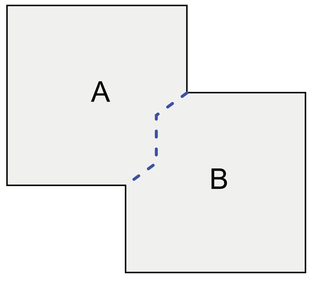
- **Divide entire extent into blocks**
- **Calculate block means per-image per-band**
- **Average block means to get reference blocks**
- **Divide image blocks with reference blocks to get local brightness correction**
- **Upscale and normalize to gain values via adaptive gamma correction**

Voronoi Center Seamline—create seamless mosaic polygons



- **Align image grids**
- **Extract valid pixel polygon**
- **Calculate polygon overlaps**
- **Build Voronoi diagrams within overlaps**
- **Extract shortest path between edge intercepts**
- **Cut image polygons by shortest paths**

Mask and Merge—composite images into a mosaic

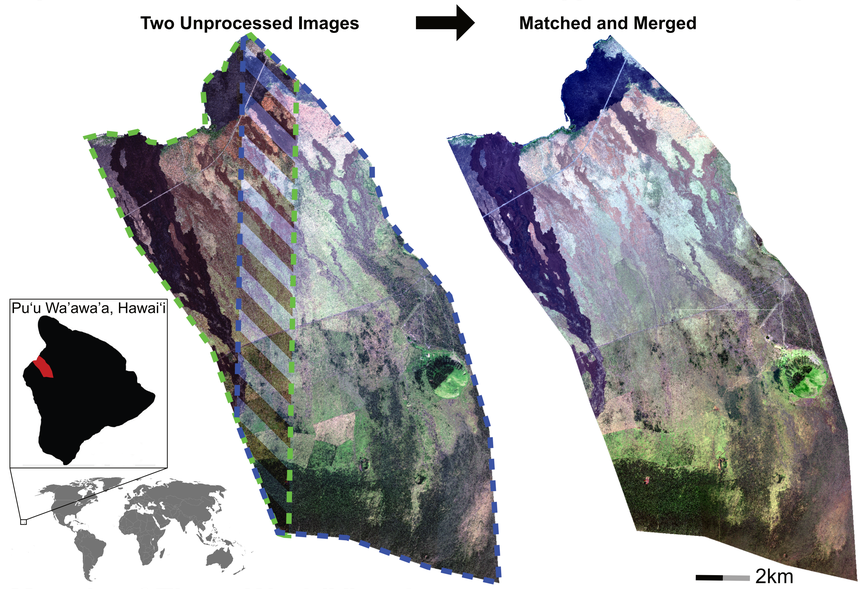


- **Clip images by the Voronoi polygons**
- **Merge images**
- The final composite is **radiometrically consistent** across scenes, **locally refined** to minimize seamline artifacts, and assembled into a center-weighted **mosaic**.

Examples

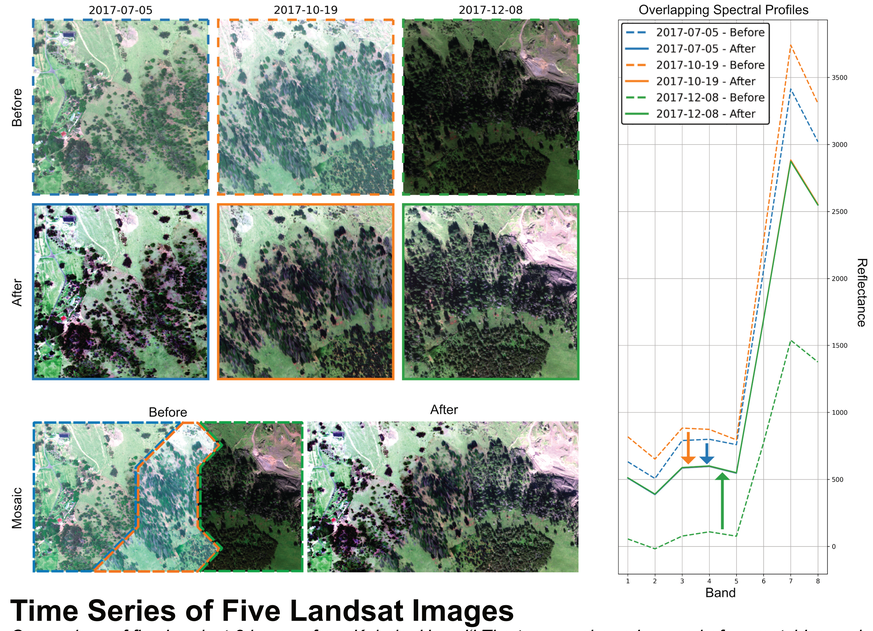
Large Scale WorldView Mosaic of Pu'u Wa'awa'a, Hawai'i

Large scale mosaic of 162,873,479 ha in Pu'u Wa'awa'a, Hawai'i using global, local, and Voronoi algorithms.



Mosaic of Three WorldView Images

Comparison of three WorldView-3 images from Pu'u Wa'awa'a, Hawai'i. Top left shows images before processing, the middle left shows images after processing, the bottom left shows images mosaicked before and after processing, and lastly, the right shows the spectral profile of all images.



Time Series of Five Landsat Images

Comparison of five Landsat-8 images from Kohala, Hawai'i.The top row shows images before matching and the bottom shows images after applying OmniCloudMask and global and local algorithms with custom Pseudo-Invariant Features.