Submerged Aquatic Vegetation Mapping using Object-Based Image Analysis with Lidar and RGB Imagery

Victoria Price

Version 1, April 16 2015

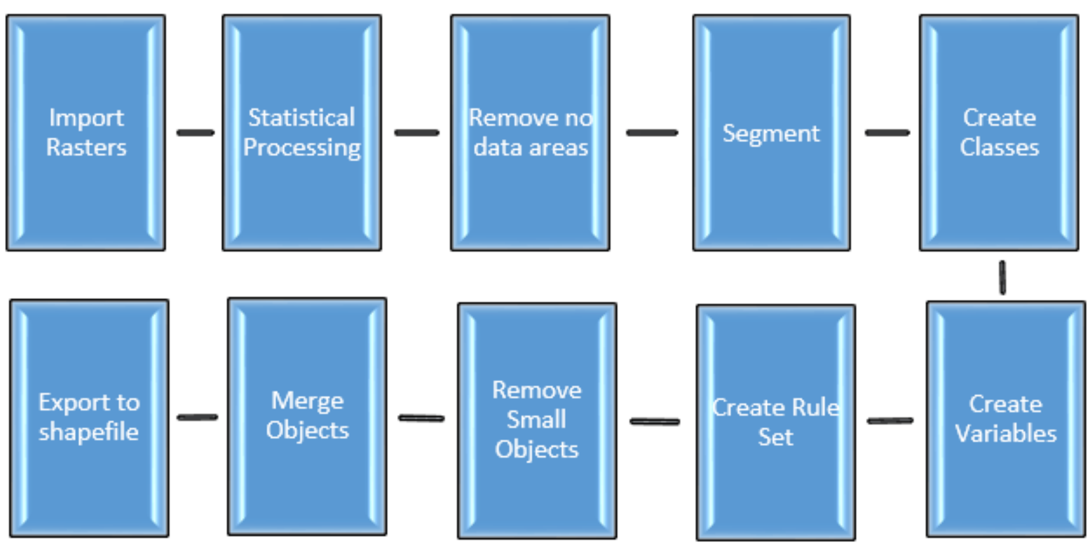
Submerged Aquatic Vegetation Mapping using Object-Based Image Analysis with Lidar and RGB Imagery

Introduction:

Traditional methods of classifying submerged aquatic vegetation (SAV) involve manual delineation in ArcGIS. This methodology is not only time consuming but is also very subjective, particularly when multiple people are analyzing long-term data sets. Additionally, these methods focus solely on classifying habitat using aerial imagery, which can vary in quality across years. Topobathy lidar, frequently collected in response to major weather events, provides additional data layers that allow for a more comprehensive data set from which to classify SAV. These layers include bathymetry and reflectance, and methodology development for the extraction and use of various additional waveform features is ongoing.

This guide will provide an overview for the semi-supervised classification of SAV using object-based image analysis (OBIA) to integrate lidar and imagery data. This methodology not only greatly reduces the time required to create habitat maps, but also removes a component of human error if multiple personnel analyze the data. This methodology is currently being assessed for its effectiveness in classifying SAV across data collected by multiple sensors, and development of this guide will be ongoing. The procedures outlined in this guide serve as a starting point for analysis and assumes a level of proficiency in the software used; specifics in each step will change with various data sets and with the specific OBIA software package.

Work Flow



Procedures

1. **Aggregate Data**

   Compile rasters (tiffs are the easiest to work with) of all data layers into one folder. It is helpful if the rasters are clipped in ArcGIS beforehand so that only areas with 100% overlap between layers are included.

   The layers currently used are:

   -RGB Imagry (one layer for each band)

   -Bathymetry

   -Reflectance

   Keep rasters in this order when importing; most OBIA packages use a layer hierarchy in analysis or have the option to do so.

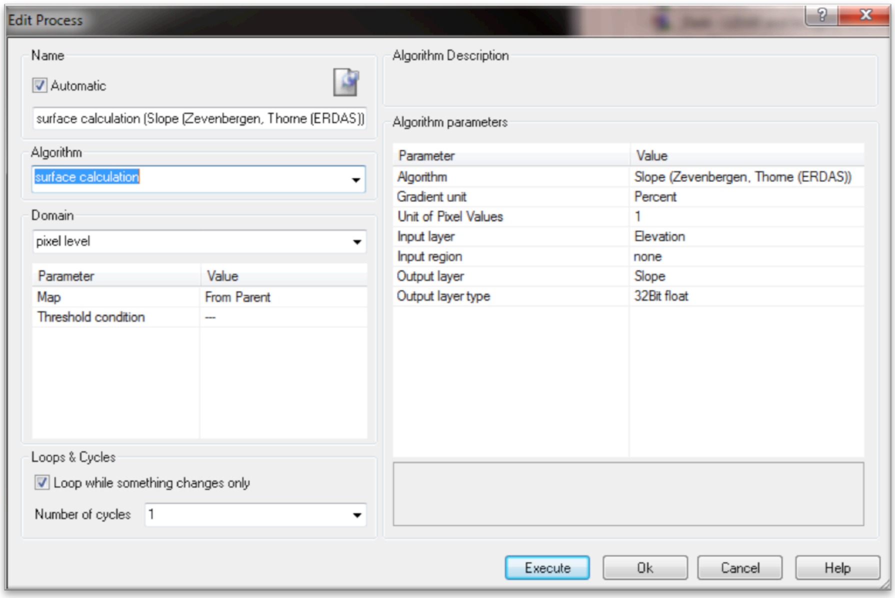
RGB Imagery

Bathymetry

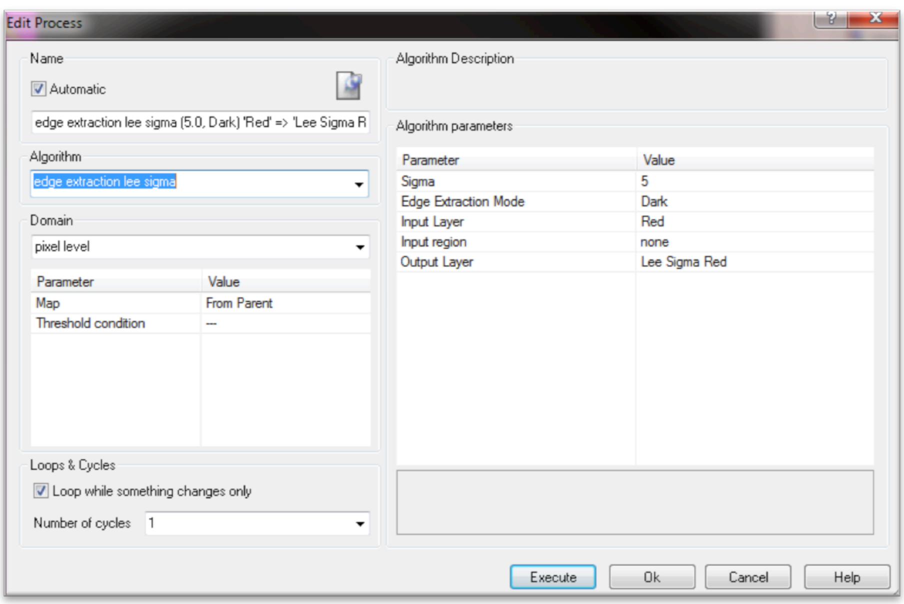
Reflectance

2. **Statistical Processing of Layers**

   First, extract the slope from the bathymetry layer. This is done using algorithms included in OBIA software packages, usually as part of the rule set that is created for classification. The gradient unit should be "percent" and the process should be performed on the pixel level.



Edge extraction follows, using the "Lee Sigma" statistical algorithm. Perform edge extraction on the reflectance and red colorband layers. The process should be performed at the pixel level, with a sigma value of 5.

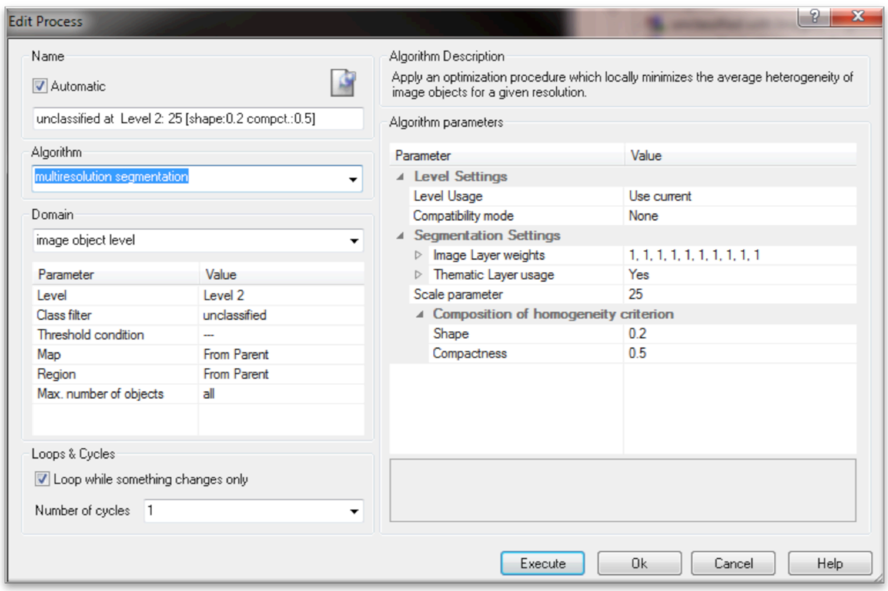


3. Remove no data/low elevation areas

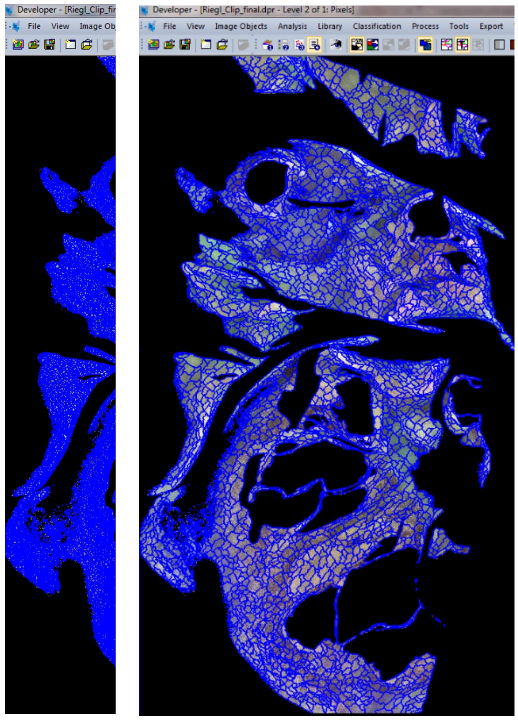
This can be done in ArcGIS or whichever OBIA package is being used. In eCognition, this is done by threshold segmentation, with low elevation areas where lidar bottom detection is nonexistent or unreliable being moved into a newly-created "Low Elevation" class and masked out. With the Reigl data set (in ellipsoid height), the threshold is -4 m, for example. The low elevation areas are then blocked out, preventing segmentation and classification and reducing processing time.

4. Segment

Segment the data set integrating all data layers equally. Choose a moderately sized segment for the first step; the number will vary greatly depending on the resolution of the data set. Smaller numbers produce smaller segments. Choose a low value for shape; higher values allow for an increased influence of color on the segmentation process. Choose a moderate value for compactness; higher values create more compact objects from segmentation.



In addition to layer and class hierarchies, there are analysis level hierarchies. Create a new level by copying the newly segmented level; ensure that it is “below” in the level hierarchy. Segment again, this time creating small segments (again, this number will vary based on the resolution of the data. For the Riegl data, it was 25).



Level 2 → Level 1

5. Create Classes

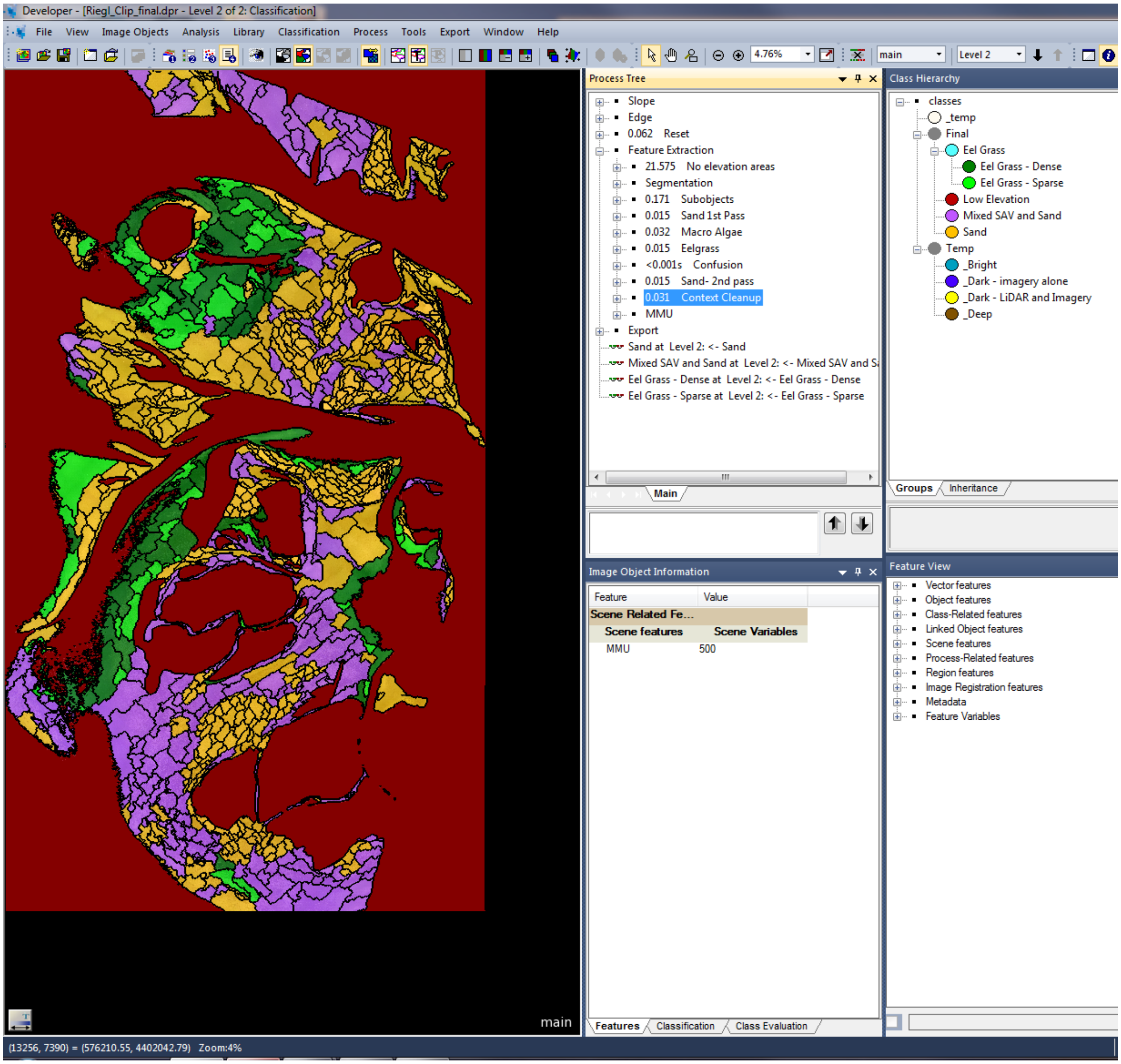
There are “temp” and “permanent” classes. Temp classes are used to provide additional information for the creation of your permanent classes.

The “temp” classes used for this classification are:

- -Bright
- -Dark in lidar and imagery
- -Dark in imagery alone
- -Deep

The permanent classes are the habitat classes and may change depending on what the end user goal is. For this classification, the permanent classes are:

- -Sand
- -Mixed SAV and Sand
- -Sparse Eel Grass
- -Dense Eel Grass



6. Create Variables

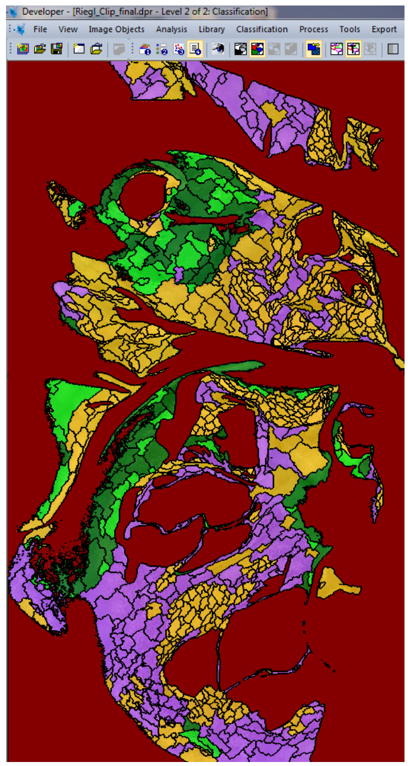
Variables are used as classification criteria and/or object properties. For habitat classification, the most useful variable is "Imagery Brightness", which is the average pixel value of all three color bands: (R+G+B)/3. It is also useful to set a minimum mapping unit (MMU); this allows for the removal of very small objects. The MMU will depend on the resolution and scale of your data.

7. Create Rule Set

Create the classification scheme or rule set. Start with level 1 (the level with small segments) and classify areas for the Temp classes: areas that are bright (have a high reflectance or imagery brightness), dark in imagery (low imagery brightness, high reflectance), dark in lidar and imagery (low imagery brightness and low reflectance), and deep areas.

Using the Temp classes, classify level 2. It is helpful to classify the most easily identifiable areas first, particularly sand. It is also helpful to use "_temp" classes (not to be confused with Temp), to classify areas of uncertainty- these objects can then be reclassified into a permanent class.

After the initial permanent classes are created, create rules that eliminate confusion and check for error (i.e. if certain objects are consistently classified as Mixed SAV and Sand when they are in fact deep sand).

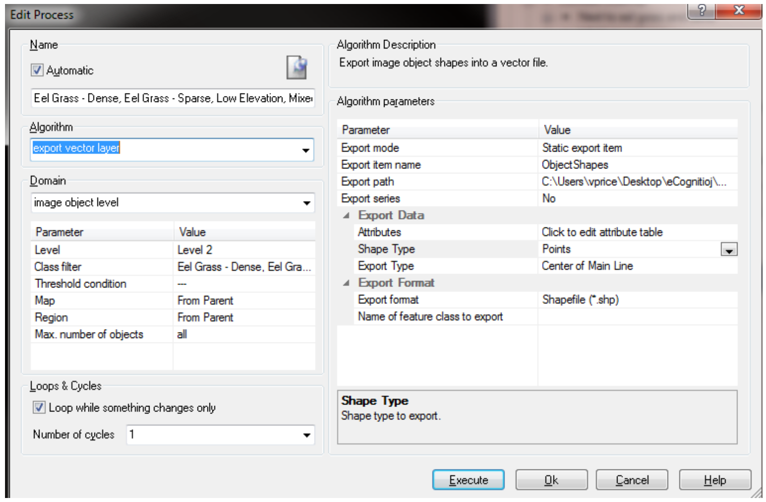


8. Remove Small Objects, Merge Objects, and Export

Remove objects with areas smaller than MMU. This removes tiny objects that are likely due to imagery irregularities.

Merge like-classed objects into one, so that one local area of Sand is not composed of multiple segments. This aids in exporting as well as creating a clear habitat map.

Export the permanent classes as a shapefile, with the shape type "polygon." Be sure to select "class name" as the object attribute to export, though other attributes such as area may be helpful in the future.


Edit Process
Name
Automatic
Eel Grass - Dense, Eel Grass - Sparse, Low Elevation, Mixe
Algorithm
export vector layer
Domain
image object level
Parameter
Value
Level
Level 2
Class filter
Eel Grass - Dense, Eel Gra...
Threshold condition

Map
From Parent
Region
From Parent
Max. number of objects
all
Loops & Cycles
Loop while something changes only
Number of cycles
1
Algorithm Description
Export image object shapes into a vector file.
Algorithm parameters
Parameter
Value
Export mode
Static export item
Export item name
ObjectShapes
Export path
C:\Users\vprice\Desktop\eCognitio\...
Export series
No
Export Data
Attributes
Click to edit attribute table
Shape Type
Points
Export Type
Center of Main Line
Export Format
Export format
Shapefile (*.shp)
Name of feature class to export
Shape Type
Shape type to export.
Execute
Ok
Cancel
Help


0
1.25
2.5
5 Kilometers
N
Legend
Riegl
Class_name
Eel Grass - Dense
Eel Grass - Sparse
Mixed SAV and Sand
Sand