Object-Based Classification & eCognition

Zutao Ouyang

11/17/2015

What is Object-Based Classification

- The object based image analysis approach delineates segments of homogeneous image areas (i.e., objects)

- In a next step, the delineated segments are classified to real world objects based on spectral, textural, neighbourhood and object specific shape parameters

Object-Based Image Analysis

- Builds on the human process of object recognition
- Replicate human interpretation of RS images in automated/semi-automated way

Why object-based classification?

Pixel based classification	Object based classification
Only based on pixel value of spectral value	Based on spectral value, and shape, texture, and context information
Ignore spatial autocorrelation	Use spatial autocorrelation
Salt-and-Pepper phenomena	None Salt-and-Pepper
	Classification process is rather fast because objects not individual pixels are assigned to specific classes

Why object-based classification?

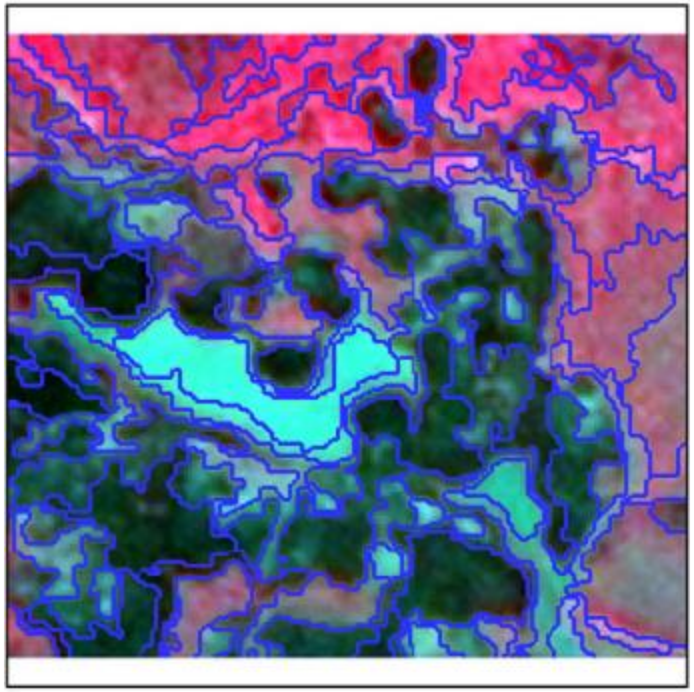
- Per-pixel classification-not suitable for processing of VHR images
  - Information content of the imagery increases with spatial resolution,
  - The accuracy of land use classification may decrease, due to increasing of the within class variability inherent in a more detailed, higher spatial resolution data

- Object-based Classification-better for VHR images especially urban ecosystems
  - Using spatial autocorrelation aggregation to reduce salts and peppers
  - Unambiguous boundaries and uniform shapes, unique textures

Image_segmentation

- Image segmentation– the process of partitioning of an image into image objects
- An image object is group of connected pixels in a scene

Before Segmentation



After Segmentation

Image Segmentation

- Algorithms

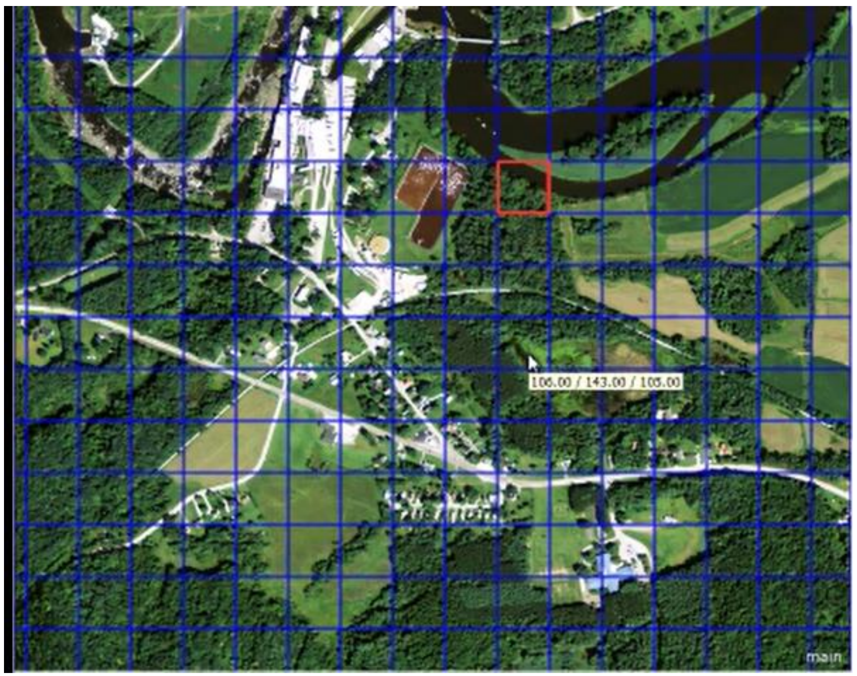
Bottom-Up Algorithms	Top-Down Algorithms

eCognition:

Bottom-Up	Top-Down
Chessboard Segmentation	Contrast Split Segmentation
Quadtree-Based Segmentation	Spectral Difference Segmentation
Multiresolution Segmentation	Multi-Threshold Segmentation

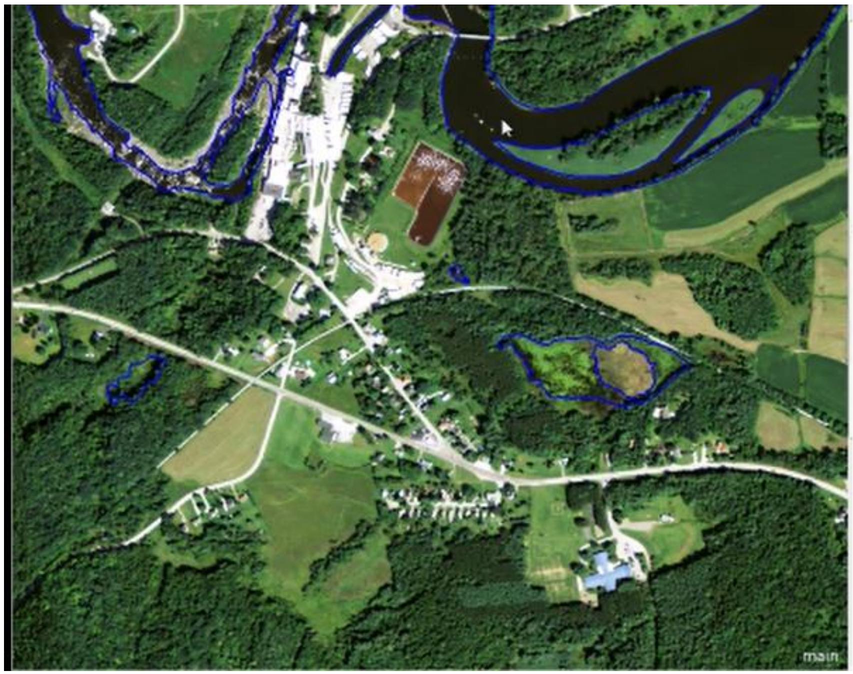
Chessboard Segmentation

Split the pixel domain or an image object domain into square image objects.



Useless to represent meaningful objects

Water body vector layer



Used to incorporate an exiting thematic layer

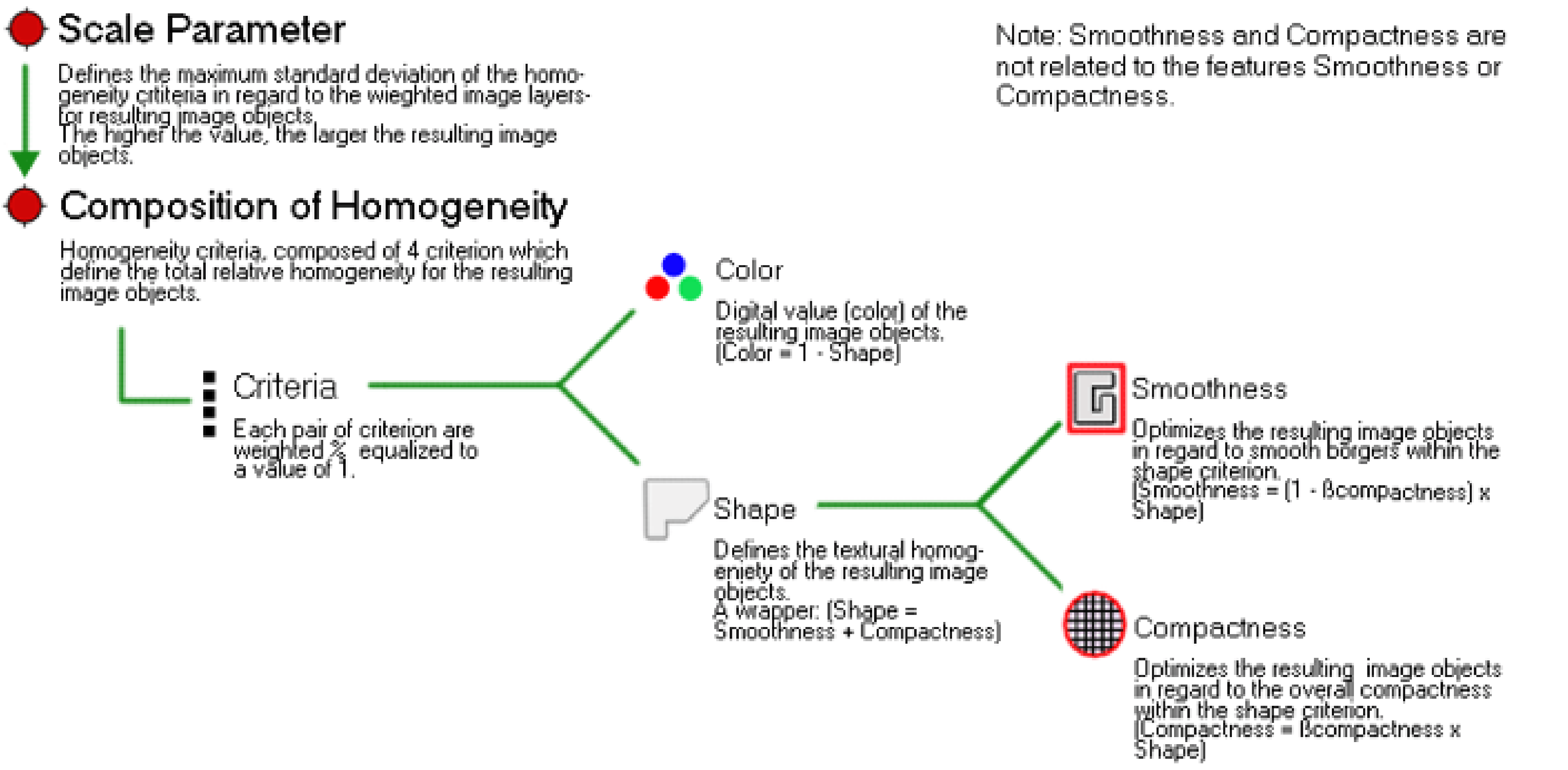
Quadtree-Based Segmentation

The Quadtree-Based Segmentation algorithm splits the pixel domain or an image object domain into a quadtree grid formed by square objects

Spectral Difference Algorithms

Neighboring image objects are merged if the spectral difference is below the value given by the maximum spectral difference.

Multiresolution Segmentation



Multiresolution Segmentation

- Scale: an abstract value to determine the maximum possible change of heterogeneity caused by fusing several objects.
  - Indirectly related to the size of the created objects.
  - At a given scale: Homogeneous areas result in larger objects, and heterogeneous areas result in larger objects.
  - Small scale number results small objects, lager scale number results in larger objects.

- Color: the pixel value

- Shape: includes compactness and smoothness which are two geometric features that can be used as "evidence."
  - Smoothness - describes the similarity between the image object borders and a perfect square.
  - Compactness- describes the "closeness" of pixels clustered in a object by comparing it to a circle

Color and Shape Criteria

These two criteria are used to create image objects of relatively homogeneous pixels using the general *segmentation function* (S_f):

$$S_f = w_{color} \cdot h_{color} + (1 - w_{color}) \cdot h_{shape}$$

where the user-defined weight for spectral color versus shape is $0 \leq w_{color} \leq 1$.

.

Spectral (i.e., *color*) *heterogeneity* (h) of an *image object* is computed as the sum of the standard deviations of spectral values of each layer (σ_k) (i.e., band) multiplied by the weights for each layer (w_k):

$$h = \sum_{k=1}^{m} w_k \cdot \sigma_k$$

Usually equal weight for all bands except you know certain band is really important

The color criterion is computed as the weighted mean of all changes in standard deviation for each band k of the m bands of remote sensing dataset. The standard deviation σ_k are weighted by the object sizes n_{ob} (i.e. the number of pixels):

$$h_{color} = \sum_{k=1}^{m} w_k \left[n_{mg} \cdot \sigma_k^{mg} - \left(n_{ob1} \cdot \sigma_k^{ob1} + n_{ob2} \cdot \sigma_k^{ob2} \right) \right]$$

where *mg* means merge.

compactness $$cpt = \frac{l}{\sqrt{n}}$$

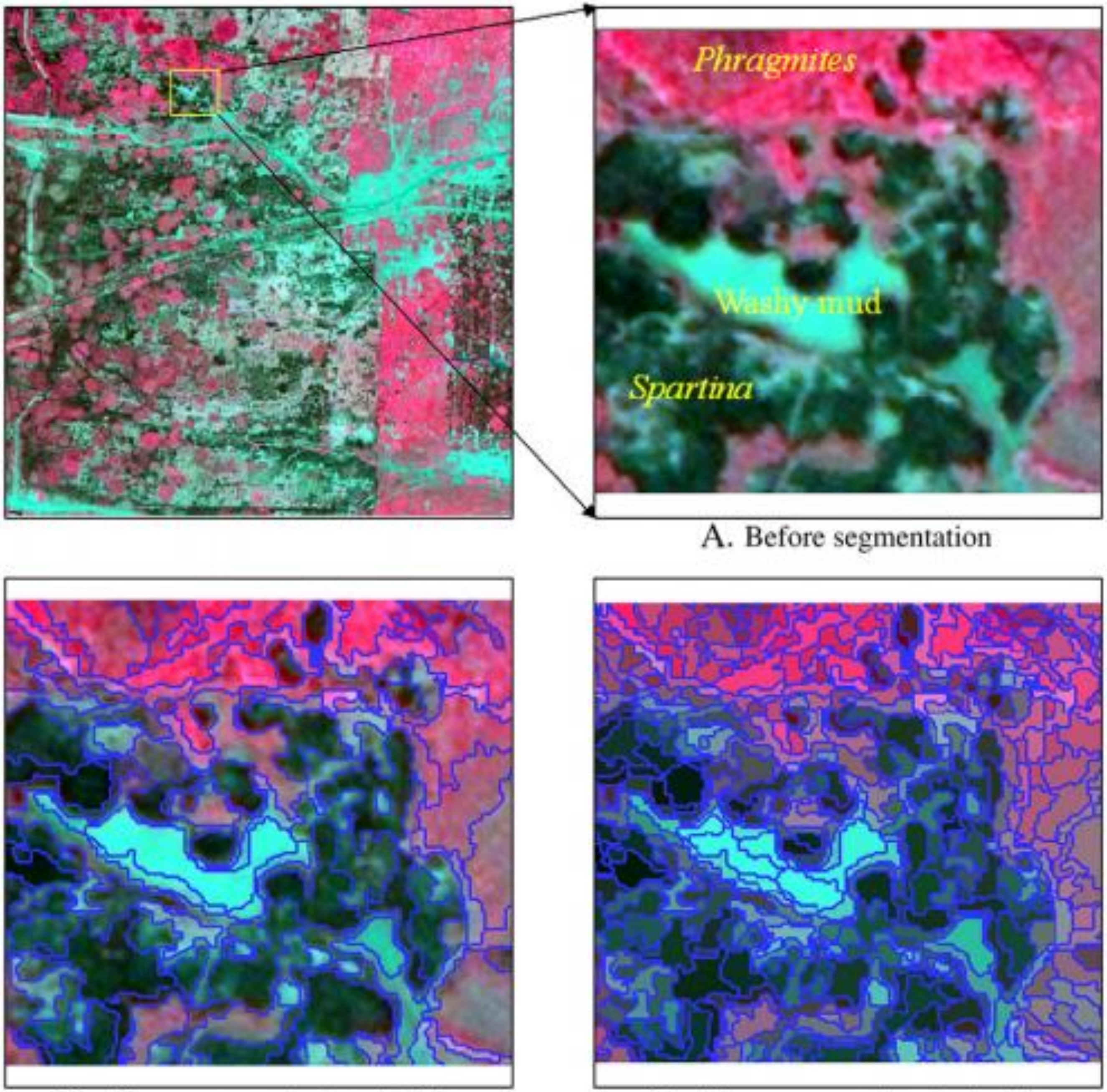
smoothness $$smooth = \frac{l}{b}$$

n is number of pixel in the object, l is the perimeter,
b is shortest possible border length of a box bounding the object

$$h_{cpt} = n_{mg} \cdot \frac{l_{mg}}{\sqrt{n_{mg}}} - \left(n_{ob1} \cdot \frac{l_{ob1}}{\sqrt{n_{ob1}}} + n_{ob2} \cdot \frac{l_{ob2}}{\sqrt{n_{ob2}}} \right)$$

$$h_{smooth} = n_{mg} \cdot \frac{l_{mg}}{b_{mg}} - \left(n_{ob1} \cdot \frac{l_{ob1}}{b_{ob1}} + n_{ob2} \cdot \frac{l_{ob2}}{b_{ob2}} \right)$$

$$h_{shape} = w_{cpt} \cdot h_{cpt} + (1 - w_{cpt}) \cdot h_{smooth}$$



A. Before segmentation

B. After segmentation(scale 35)

C. After segmentation(scale 15)

Multiresolution segmentation

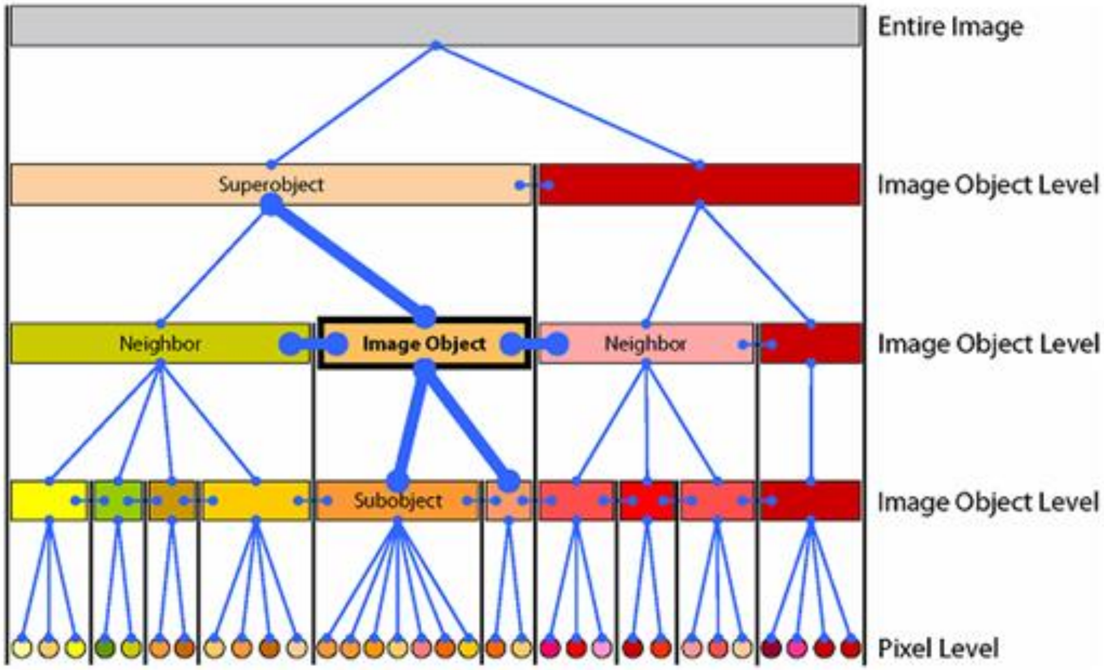
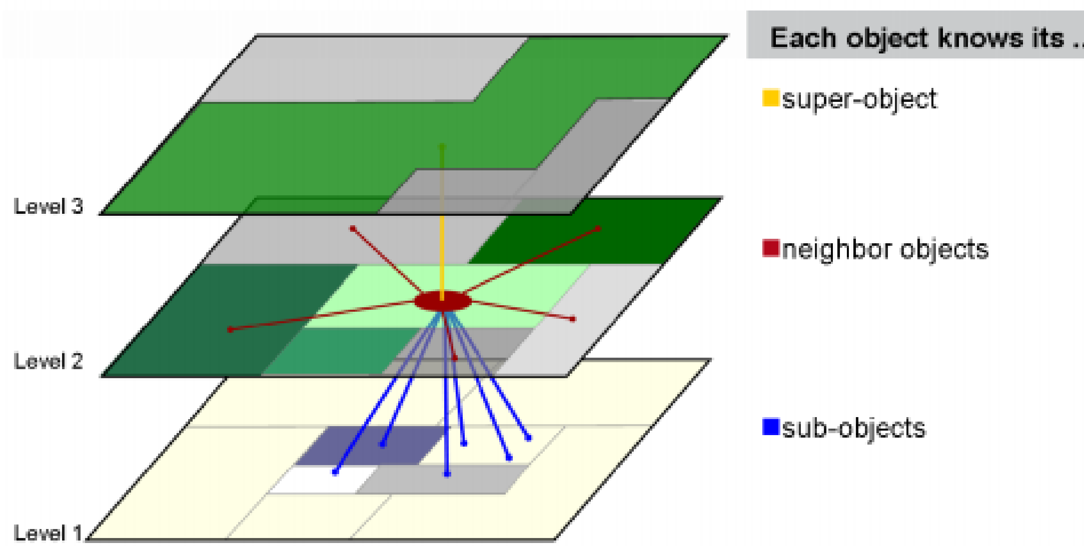


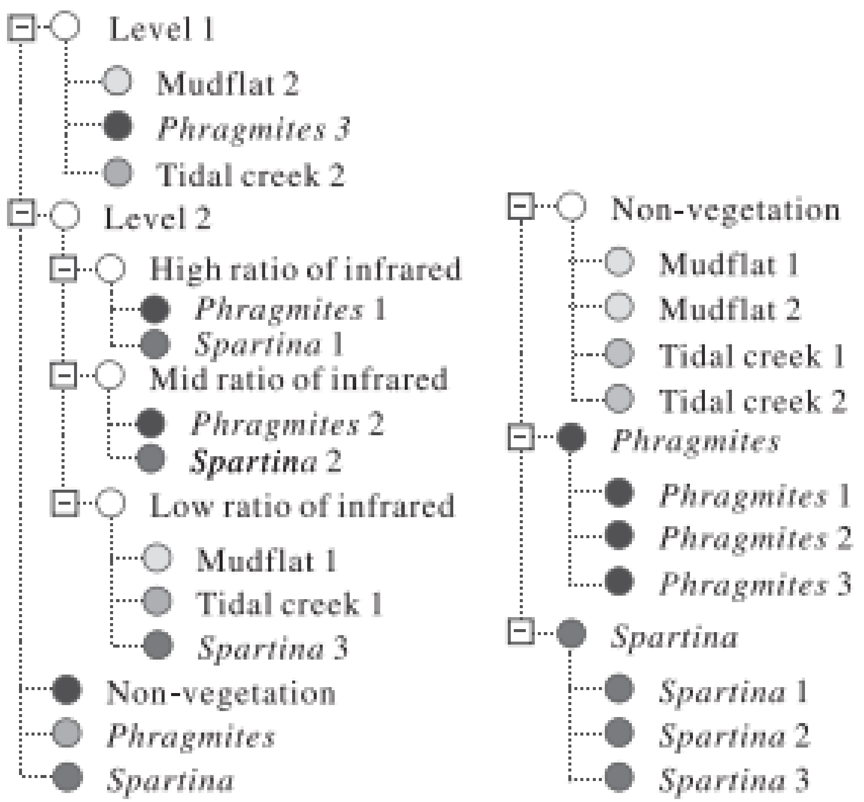
Image Object
Hierarchy

Contextual Information
Shape
Texture

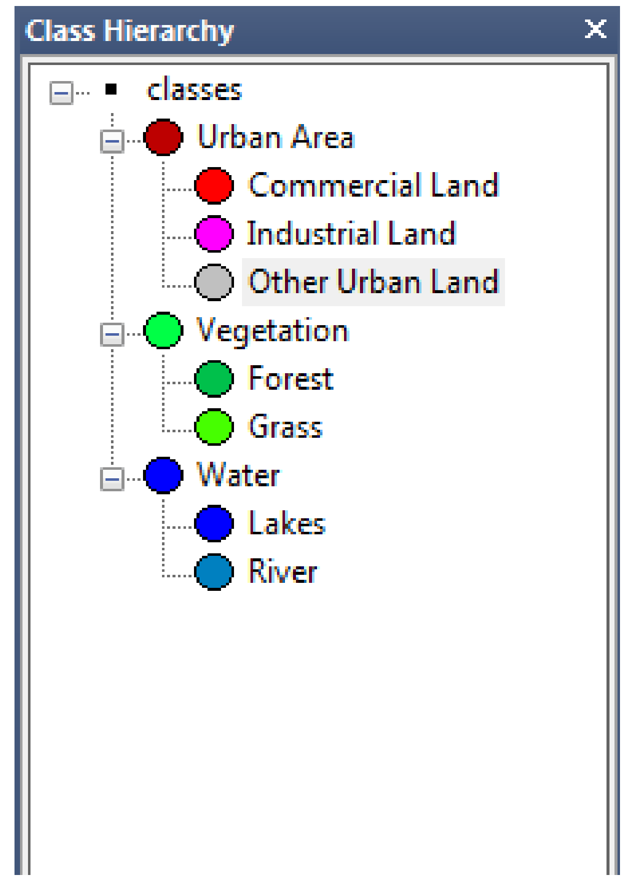


Classification

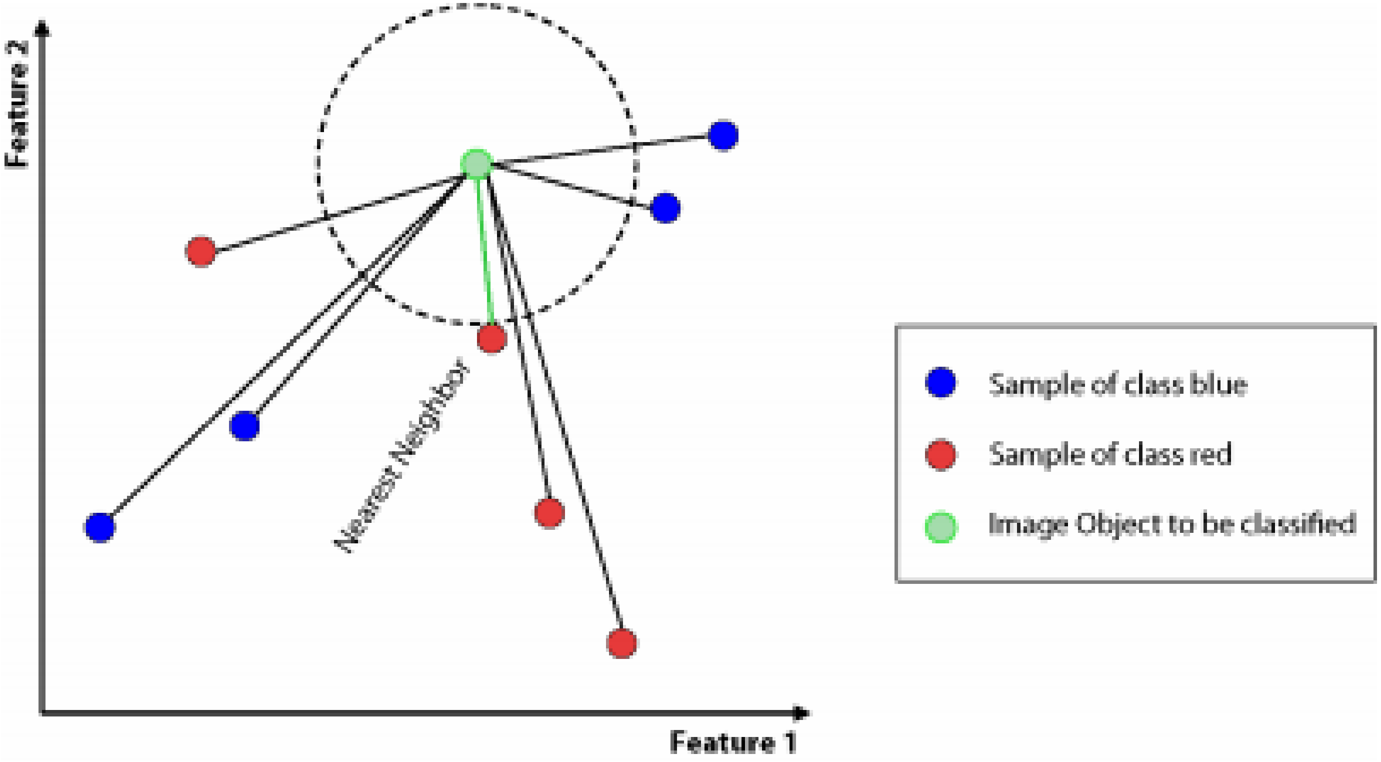
- Define Classes and Class hierarchy



Inheritance hierarchy **Group hierarchy**



Nearest Neighbor



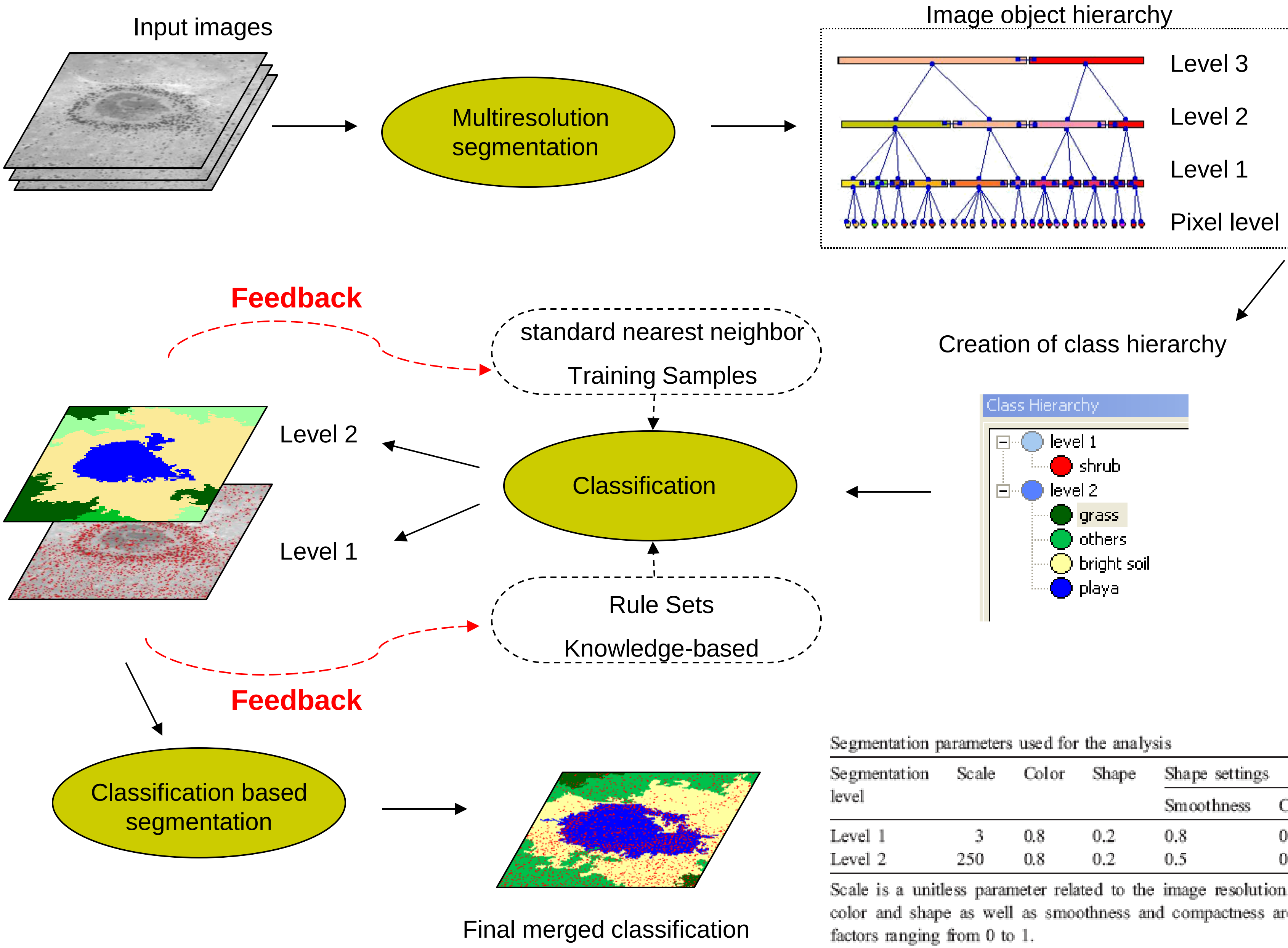
Principle of Nearest Neighbor

Rule Sets:
Assign Objects to Classes based on you expertise/prior knowledge

Assign all objects with height large than a certain value to be buildings

An iterative process to refine the results

Workflow in eCognition



Segmentation parameters used for the analysis

Segmentation level	Scale	Color	Shape	Shape settings	
				Smoothness	Compactness
Level 1	3	0.8	0.2	0.8	0.2
Level 2	250	0.8	0.2	0.5	0.5

Scale is a unitless parameter related to the image resolution. Values for color and shape as well as smoothness and compactness are weighting factors ranging from 0 to 1.