INTERIMAGE
interpreting images freely

# InterImage 1.41 User Guide

**www.lvc.ele.puc-rio.br/projects/interimage**

# Table of Contents

**1 Introduction ........................................ 1**

**2 Basic Concepts ........................................ 2**

2.1 Semantic Net ........................................ 3

2.2 Top-Down Operators ........................................ 3

2.3 Bottom-Up Operators ........................................ 4

2.4 Decision Rules ........................................ 5

2.5 Interpretation Control ........................................ 8

**3 System Interface ........................................ 11**

3.1 Main Window ........................................ 11

3.1.1 Menus ........................................ 11

3.1.2 Toolbar ........................................ 14

3.1.3 Semantic Net Window ........................................ 15

3.1.4 Layers Window ........................................ 16

3.1.5 Node Editor Window ........................................ 21

3.1.6 Viewer ........................................ 22

3.1.7 Object Information Window ........................................ 24

3.2 New/Edit Project Window ........................................ 25

3.2.1 Supported Resource Formats ........................................ 26

3.2.2 Maximum Image Size ........................................ 26

3.2.3 Resources in Different Resolutions ........................................ 27

3.3 Decision Rule Window ........................................ 27

3.3.1 Building Blocks ........................................ 28

3.3.2 Toolbar ........................................ 29

3.3.3 Upper/Lower Level Rule ........................................ 30

3.3.4 Decision Tree Tab ........................................ 30

3.3.5 Source Code Tab....32

3.3.6 Insert/Edit Class Window....32

3.3.7 Insert/Edit Selection Window....33

3.3.8 Insert/Edit Expression Window....33

3.3.9 Insert/Edit Membership Window....34

3.3.10 Membership Function Window....35

3.3.11 Insert/Edit Aggregation Window....37

3.4 Analysis Explorer Window....38

3.4.1 Control Panel....39

3.4.2 Analysis Tools....40

3.5 Shapefile Editor Window....41

3.6 Samples Editor Window....42

**4 Batch Processing....46**

**References....47**

# 1 Introduction

InterImage is an open source software development initiative that is part of an international scientific cooperation project led by the Computer Vision Laboratory of the Department of Electrical Engineering of the Catholic University of Rio de Janeiro (PUC-Rio) and by the Image Processing and Remote Sensing divisions of the National Institute for Space Research (INPE).

InterImage is a multi-platform system for image automatic interpretation written in C++ and Qt. The system provides support for the integration of external image processing operators that can be coded in any programming language or even be proprietary programs. In its basic package, InterImage offers, however, a set of operators constructed with the functions and classes provided by TerraLib [1] called TerraAIDA (http://www.dpi.inpe.br/terraaida) and with the routines provided by the BREC [2] Library (http://tolomeofp7.unipv.it/SoftwareTools/BREC).

InterImage is based on the GeoAIDA system [3], developed by the Institute of Technology Information of the University of Hannover [4], Germany, and inherited from this system its basic functional characteristics, besides knowledge structures and control mechanisms. A new graphical user interface, knowledge representation functionality and image processing operators were later added to the system.

Chapter 2 of this manual will present the basics concepts of the system and some theoretical foundations that will help in the understanding of its operation. The system interface will be presented in Chapter 3, along with its main features and screens. For practical content as examples of interpretation projects, tutorials etc. visit our wiki (http://wiki.dpi.inpe.br/doku.php?id=interimage). Reports about problems, requests for additional information and suggestions about new features can be sent to lvc_inter@ele.puc-rio.br.

# 2 Basic Concepts

Figure 2.1 describes the components of the interpretation process in InterImage. The system implements a specific interpretation control strategy, guided by a structured knowledge model through a semantic net. The interpretation control (Section 2.5) is executed by the system core, which uses as input a set of geo-referenced images, SIG layers, digital elevation data or other geo-registered data. Through the interpretation of the scene, input data are processed with the help of external programs, called top-down and bottom-up operators.

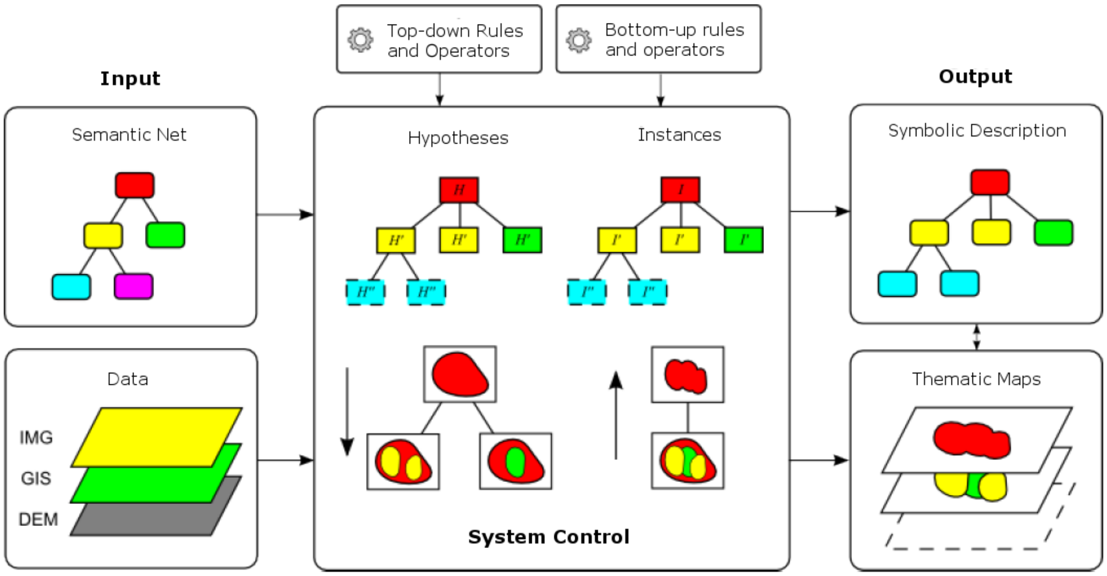


Figure 2.1 – Analysis process components.

Top-down operators are responsible for the partition of the scene into regions, considered as object hypotheses. This is a preliminary classification which identifies segments with the potential to belong to each class. The bottom-up operators refine the classifications produced in the top-down step, confirming or rejecting them and solving possible spatial conflicts between them. At the end of the interpretation process, the hypotheses become validated object instances.

The output of the interpretation process is a symbolic description of the scene, consisting mainly of a net of object instances and labeled images that correspond to regions associated with object classes. From the labeled images the system allows the creation of different thematic maps representing the different levels of semantic concepts in the net.

## 2.1 Semantic Net

A knowledge model in InterImage contains information used by the control process for the interpretation of a scene. It is represented by a semantic net (Figure 2.2), where the nodes organization is hierarchical and each node can be associated only with one ancestor node (parent) and one or more child nodes (children).

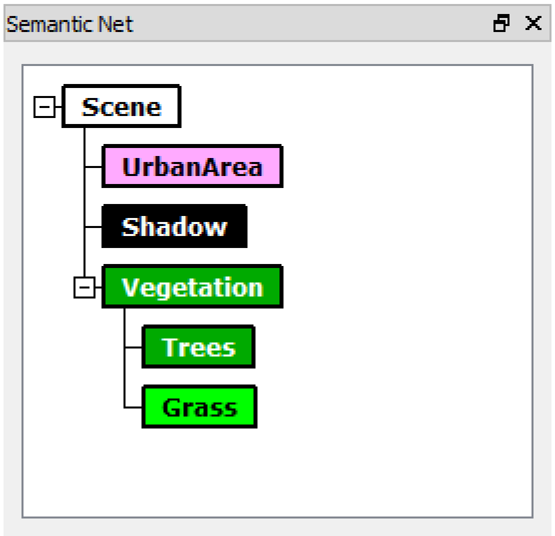


Figure 2.2 – Semantic net.

Each node in the semantic net corresponds to an object class expected to be found in the scene. Nodes have properties, such as top-down and bottom-up operators as well as generic parameters and other specific operators.

**See also**

Semantic Net window, page 15

## 2.2 Top-Down Operators

When building the interpretation model, the user attaches top-down operators to each node of the semantic net. The top-down operator task is to identify objects in the image under consideration that are likely to belong to the class corresponding to the semantic node to which it is associated.

Top-down operators are executable programs, called by the system core during the process of interpretation. They can in principle handle not only images, but also any type of geo-registered data, including vector data in a GIS database, digital elevation models or other types of raster data.

When the core calls the top-down operator, it passes to the operator information about the geographical boundaries of the region to be processed. This region of interest (ROI) is defined by another operator top-down associated to an ancestor node. Some top-down operators may associate confidence values to the hypotheses identified, which may later be used to evaluate these hypotheses by a bottom-up operator.

Decision rules (Section 2.4) can be explicitly defined by the user to post-process the objects identified by the operator. Properties of the object hypotheses can be used in this context. This processing can mean simply discard some hypotheses, calculate new confidence values or even refine the preliminary classification.

You can check a node of the semantic network as **TopDown Multi-Class**. There can be only one node of this type for the child nodes of the same parent node. Thus, the top-down operator associated to this node will be responsible for identifying objects in the image not only of the respective class, but also of sibling nodes classes. In this case, the operators associated with sibling nodes will not run. The operator associated with the multi-class node class needs to be able to identify objects from more than one class or a decision rule must be created for this purpose.

The system provides a default top-down operator called **Dummy Topdown**. This operator will output a single region that is equal to the ROI defined in the parent node of the node to which it is associated. This operator allows setting the confidence value of its output and the project image that it will be associated to. However, it doesn't allow using decision rules due to the characteristic of its processing.

**Note**

For information on other top-down operators and their parameters, visit
http://wiki.dpi.inpe.br/doku.php?id=interimage:operators documentation
http://wiki.dpi.inpe.br/doku.php?id=interimage:brec operators documentation.

**See also**

Node Editor window, page 21

## 2.3 Bottom-Up Operators

Bottom-up operators can also be associated to each node of the semantic net. The bottom-up operator processes the hypotheses of child nodes of the node to which it is associated, generated in top-down step. It can validate hypotheses and discard, or resolve spatial conflicts.

Bottom-up operators are also executable programs, called by the system core during the interpretation process. The input of such operators is a list of regions, each region is associated to an object hypothesis belonging to the classes of the child nodes.

Decision rules (Section 2.4) can be explicitly defined by the user to post-process the judgment made by the operator. Properties of the object hypotheses can be used in this context. This processing may mean discard/validate hypotheses or resolve spatial conflicts. The validated hypotheses will then be considered object instances.

It is important to note that instances of objects can, at a later stage of the interpretation process, be discarded. This will happen if a hypothesis of a higher-level object is discarded.

The operator also groups the instances of objects, assigning to each group a region equivalent to the union of the regions associated to each instance. The groups of objects will originate new hypotheses for the semantic node to which the operator is associated, replacing the original hypothesis, as will be explained in Section 2.5.

The system provides a default bottom-up operator called **Dummy Bottom-Up**. This operator performs no processing, leaving it to the decision rule to judge the hypotheses of child nodes objects.

**Note**

For information on other top-down operators and their parameters, visit
http://wiki.dpi.inpe.br/doku.php?id=interimage:operators_documentation
http://wiki.dpi.inpe.br/doku.php?id=interimage:brec_operators_documentation.

**See also**

Node Editor window, page 21

## 2.4 Decision Rules

Decision rules can be used both to reclassify object hypotheses generated by top-down operators or to decide between competing hypotheses of objects during the bottom-up step. The decision rules defined for an arbitrary node of the semantic net are always executed after the execution of the respective top-down and bottom-up operators associated to that node.

InterImage has a specific graphical user interface (Section 3.3) to support the definition of decision rules. Through this interface the user can code simple rules, whose basic elements, called building blocks are shown in Figure 2.3.

A decision rule processes and presents as output a set of objects. It can be considered that the basic steps of a decision rule are: (i) select a set of objects, (ii) filter this set of objects (discarding objects within the set), (iii) assign a degree of membership to objects within the set, and (iv) resolve spatial conflicts among objects in the set. This last step is only meaningful for decision rules associated with the bottom-up step. The steps listed above can be combined in different ways to create complex rules.

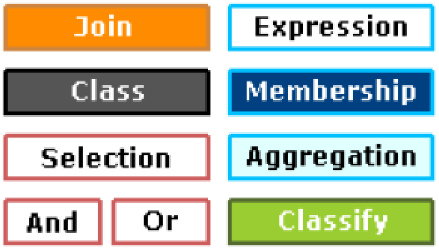


Figure 2.3 – Decision rule building blocks.

The **Class** building block allows selecting objects of a particular class (associated to a semantic node). With this block a set of objects is created which can be joined to another set through the **Join** block. Figure 2.4 shows a simple bottom-up decision rule for the **Vegetation** node of the semantic net shown in Figure 2.2. Basically what the rule does is select all the object hypotheses generated in the top-down step for **Trees** and **Grass** nodes, join these hypotheses (through **Join** block) and resolve spatial conflicts between the hypotheses of the two classes of objects (through **Classify** block, specializing in the rule to the **Spatial Resolve** block). It is interesting to note that if there is a partial spatial conflict between **Trees** and **Grass** hypotheses, the hypothesis with the lowest membership value will not be completely discarded - only the region that intersects the other hypothesis is suppressed, e.g., the region of the hypothesis with lower membership value will shrink.

In a decision rule, InterImage can calculate a variety of attributes for the hypotheses of selected objects, attributes based on spectral values, shape, texture and topological characteristics of image segments associated with those hypotheses. These attributes can be used to select objects within a set, through **Selection** block, with a user-defined threshold. In Figure 2.5 a combination of selection blocks is used to filter the set of objects created in the top-down step for class **Trees**. All objects that do not meet the selection criteria will be removed from the set.

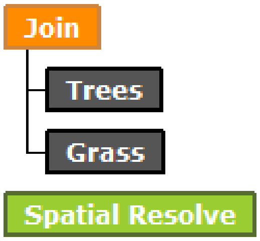
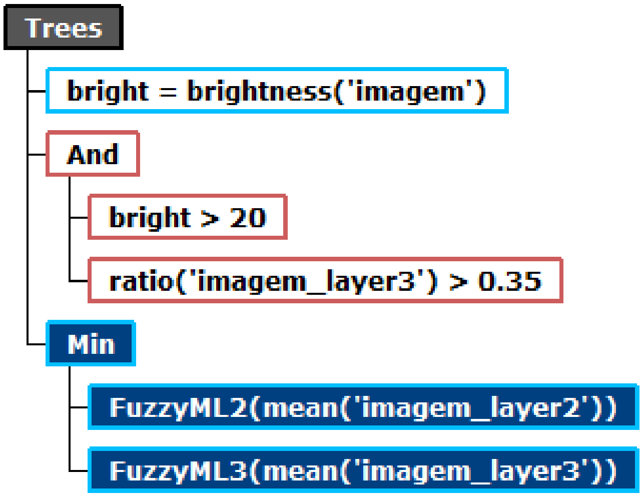


Figure 2.4 – Example of a bottom-up decision rule.

The **Expression** block allows you to create variables with user-defined names from attributes of object hypotheses. These variables are associated with each object hypothesis, e.g., for each different object it may have a distinct value. In Figure 2.5, the **Expression** block is used to store the brightness attribute value of each hypothesis. This variable is then used in a block selection. The **Expression** block allows creating complex arithmetic expressions from the attributes calculated by InterImage.

The **Membership** block allows the user to define a membership value for objects, which can be done through a combination of membership functions, as shown in Figure 2.5. The **Aggregation** block allows the aggregation of attribute values for the set of selected objects.



```
Trees
├ bright = brightness('imagem')
├ And
│ ├ bright > 20
│ └ ratio('imagem_layer3') > 0.35
└ Min
  ├ FuzzyML2(mean('imagem_layer2'))
  └ FuzzyML3(mean('imagem_layer3'))
```

Figure 2.5 – Example of a top-down decision rule.

Figure 2.5 shows an example of a simple top-down decision rule for the **Trees** node in the semantic net in Figure 2.2. In this case, a segmentation top-down operator was associated to the node **Trees**. Initially, all segments for which the brightness and the ratio of the band 4 average value are larger than certain thresholds are selected to be regarded as hypotheses of **Trees**. Then each selected hypothesis is given a value equal to the minimum value relevance between the **FuzzyML2** and **FuzzyML3** membership functions, defined respectively on the average values of the pixels that compose the segments corresponding to the bands 2 and 3 of the image. The membership functions are defined interactively by the user. The function **FuzzyML2** is shown in Figure 2.6.

The last operation of a decision rule is a union operation. This operation is responsible for spatial grouping the set of hypotheses selected at the end of the decision rule. There are three possibilities: **Merge All** - all hypotheses are combined into a single hypothesis, which can cover a not contiguous area; **Merge Connected** - each group of spatially connected hypotheses are combined into a single hypothesis, covering a contiguous region, or **No Merge** - hypotheses resulting from the decision rule are not merged.

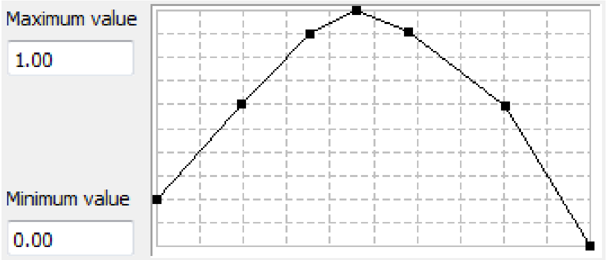


Figure 2.6 – Example of a membership function.

**See also**

Node Editor window, page 21

Decision Rule window, page 27

## 2.5 Interpretation Control

The main task of the system core is to control the interpretation process, which consists of two complementary steps: top-down and bottom-up.

In the top-down step the control process traverses the nodes of the semantic net, from the root to the leaf nodes calling top-down operators and decision rules associated to each node. The top-down processing occurs in parallel with respect to the branches of the semantic net. Object hypotheses associated to semantic nodes are created during this process and organized in a net of hypotheses. When the processing reaches the leaf nodes, it starts the bottom-up step. From there, the control process passes to visit nodes in the opposite direction, calling their bottom-up operators and decision rules recursively until the root node is reached, and that a network of instances have been created. In this case the object hypotheses are discarded or turned into object instances. A more formal description of the interpretation process is shown in Figure 2.7.

$N$ represents a generic node of a net. $N$’ is a child node of $N$. $N$’’ is child of $N$’ and so on. $T_N$ represents a hierarchically structured net, i.e., a tree, and $N'_m$ and $N'_n$ represent different nodes in the same hierarchical level of $T_N$.

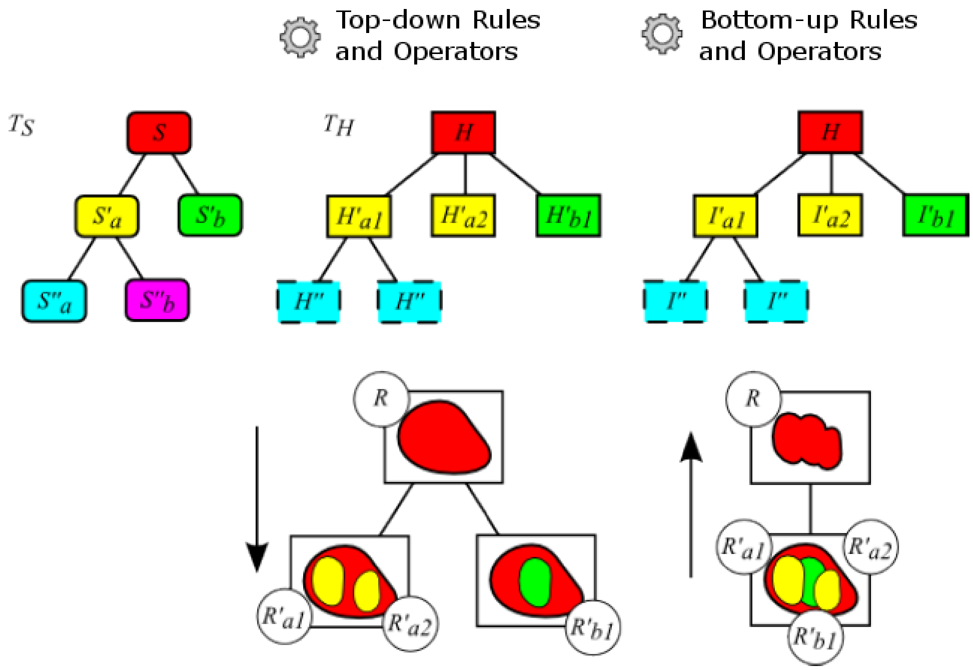


Figure 2.7 – Interpretation process flowchart.

Let $S$ be a node of the semantic net $T_S$ and let $H$ and $I$ be the nodes of the $T_H$ and $T_I$ nets, created by the interpretation process. $H$ represents the node associated to an object hypothesis and $I$ represents a node associated to an object instance.

Let $R$ be the representation of a region in the scene to which $H$ and $I$ are associated. $R'$ is therefore associated to $H'$ or $I'$. Considering the hypothesis net $T_H$, all $R'$ associated to the $H'$ nodes are subsets of $R$; considering the instance net $T_I$, all $R'_m$ and $R'_n$ pairs associated to the $I'_m$ and $I'_n$ nodes are disconnected (don’t spatially intersect).

The aim of the interpretation process is to create an instance net $T_I$ and a corresponding region net $T_R$, applying the knowledge represented by the semantic net $T_S$ in the interpretation of the region $R$. Initially, the hypothesis net $T_H$ is created and gradually hypothesis nodes $H$ are replaced by instance nodes $I$, so that in the end of the process, the $T_I$ net is complete.

The top-down processing is shown in the center of Figure 2.7. It starts (in the figure) at the point where the hypothesis node $H$, associated to the region $R$ and to the semantic node $S$, has been created. From this point on, the control passes recursively to the $S'$ nodes.

Hypothesis nodes $H'_{mi}$ are generated through the execution of a top-down operator and a decision rule (if there is one) associated to the semantic node $S'_m$, for the region $R$. The $H'_{mi}$ nodes are associated to the $R'_{mi}$ regions. If a **Dummy Top-down** operator is associated to the node $S'_m$, a simple hypothesis node, $H'_{ml}$, will be associated to all the region defined by its parent node $H$. $R'_{ml}$ in this case will be equal to $R$.

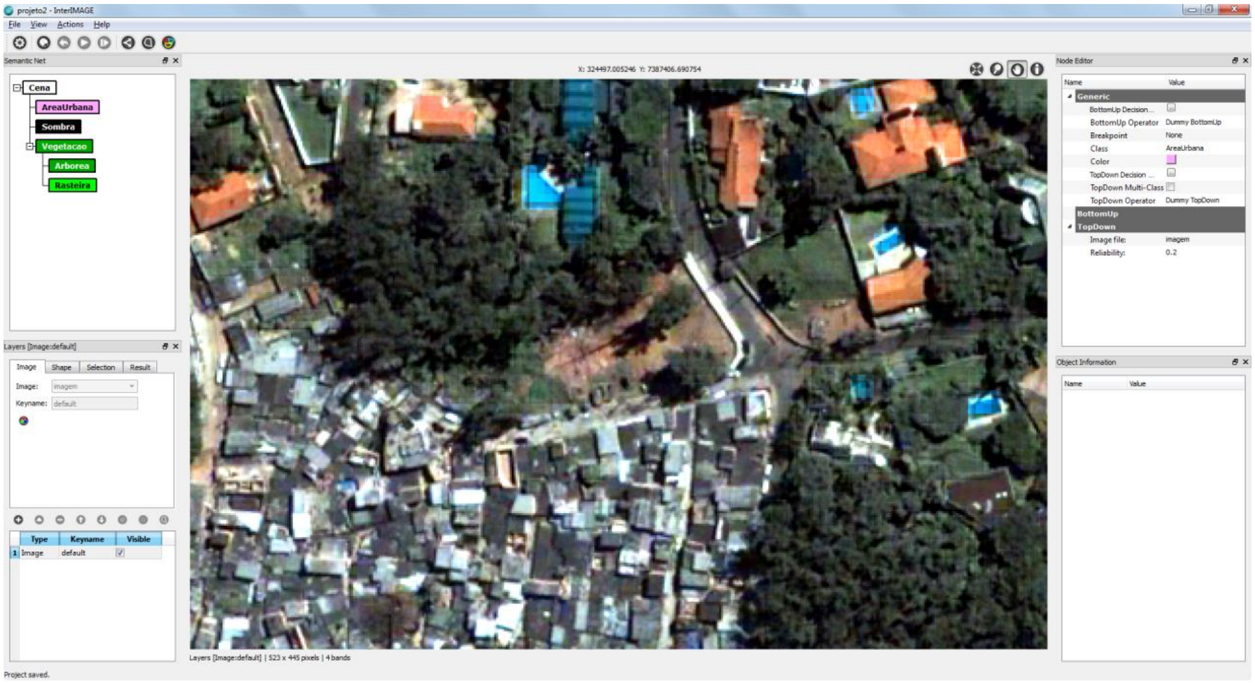
If $S'_m$ has child nodes, the procedure mentioned in the previous paragraph will be repeated for each $H'_{mi}$ until the semantic leaf nodes are reached. At this point, begins the bottom-up processing (to the right, Figure 2.7), initiating with the leaf nodes parents. The bottom-up operator and decision rule associated to $S'_m$ will be executed for $H'_m$ as soon as all $H''_{ni}$ nodes have been created - the $n$ index identifies the direct descendants of $H'_m$. The operator/rule will evaluate the hypothesis nodes $H''_{ni}$ and decide if they will turn into instance nodes $I''_n$ or will be removed from the net. In addition, the bottom-up operator/rule will turn the regions $R''_{ni}$ associated to $I''_{ni}$ nodes disjoint (solving eventual spatial conflicts).

The bottom-up operator/decision rule will group the $I''_n$ nodes and will generate new hypothesis nodes $H'_g$ to which the instance nodes will be connected. $H'_g$ will, then, be put in the hypothesis net and associated to node $H$, being the original node $H'_m$ removed. The control is then passed to node $H$, for evaluation and grouping of nodes $H'$. The bottom-up continues until the hypothesis net root node is processed. At this point, the instance net $T_I$ will be complete.

# 3 System Interface

This chapter will guide you through the main elements of the system interface such as: menus, toolbars, dialog boxes and windows.

## 3.1 Main Window



X: 324497.005246 Y: 7387406.690754

Image: imagem

Keyname: default

| Name | Value |
|---|---|
| **Generic** | |
| BottomUp Decision... | |
| BottomUp Operator | Dummy BottomUp |
| Breakpoint | None |
| Class | AreaUrbana |
| Color | 🟪 |
| TopDown Decision ... | |
| TopDown Multi-Class | □ |
| TopDown Operator | Dummy TopDown |
| **BottomUp** | |
| **TopDown** | |
| Image file: | imagem |
| Reliability: | 0.2 |

| | Type | Keyname | Visible |
|---|---|---|---|
| 1 | Image | default | ☑ |

Layers [Image:default] | 523 x 445 pixels | 4 bands

Project saved.

Figure 3.1 – Main window.

The elements of the main window (Figure 3.1) are:

### 3.1.1 Menus

### 3.1.1.1 File Menu

The **File** menu (Figure 3.2) provides the following options:

**New Project –** Creates an interpretation project.

**Open Project –** Opens a project.

**Edit Project –** Edits the current project.

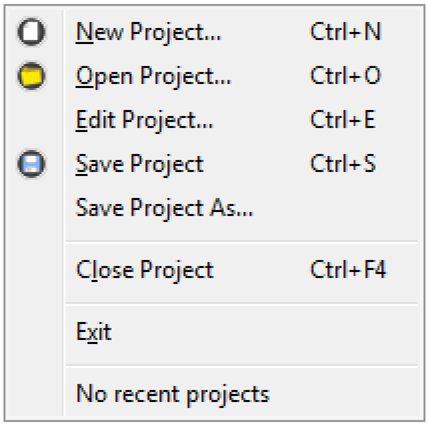


Figure 3.2 – **File** menu.

**Save Project** – Saves the current project.

**Close Project** – Closes the current project.

**Exit** – Quits the program.

Below the **Exit** option is offered a list of recent projects, making it easy to return to a previous project in which you were working on.

**See also**

New/Edit Project window, page 25

### 3.1.1.2 View Menu

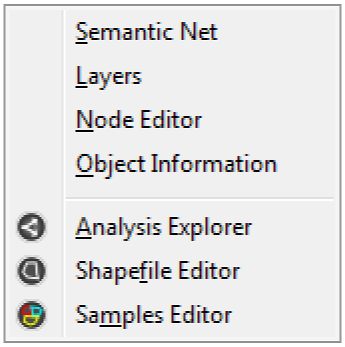


Figure 3.3 – **View** menu.

The **View** menu (Figure 3.3) provides the following options:

**Semantic Net** – Displays the **Semantic Net** window.

**Layers** – Displays the **Layers** window.

**Node Editor –** Displays the **Node Editor** window.

**Object Information** – Displays the **Object Information** window.

**Analysis Explorer –** Tool that helps in building the interpretation model.

**Shapefile Editor** – Tool for creating and editing shapefiles.

**Samples Editor –** Tool for segmentation, samples collection and manual classification of polygons.

**See also**

Analysis Explorer window, page 38

Shapefile Editor window, page 41

Samples Editor window, page 42

## 3.1.1.3 Actions Menu

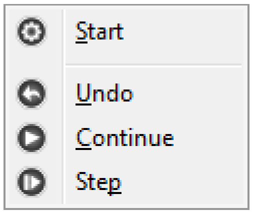


Figure 3.4 – **Actions** menu.

The **Actions** menu (Figure 3.4) provides the following options:

**Start** – Executes an interpretation.

**Undo** – Undoes the previous interpretation step.

**Continue** – Continues the interpretation process until find another breakpoint. If this does not exist, performs the interpretation until the end.

**Step** – Goes one step ahead in the interpretation.

### 3.1.1.4 Help Menu

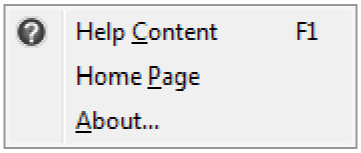


Figure 3.5 – **Help** menu.

The Help menu (Figure 3.5) provides the following options:

**Help Content** – Opens the wiki page where part of the program documentation is concentrated.

**Home Page** – Opens InterImage website.

**About** – Displays information about the program.

## 3.1.2 Toolbar

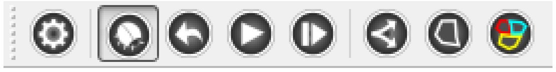


Figure 3.6 – Toolbar.

The toolbar (Figure 3.6) provides the following options:

**Execute** – Executes an interpretation.

**Debug Mode** – Enables/disables debug mode.

**Undo** – Undoes the previous interpretation step.

**Continue** – Continues the interpretation process until find another breakpoint. If this does not exist, performs the interpretation until the end.

**Step** – Goes one step ahead in the interpretation.

**Analysis Explorer** – Tool that helps in building the interpretation model.

**Shapefile Editor** – Tool for creating and editing shapefiles.

**Samples Editor** – Tool for segmentation, samples collection and manual classification of polygons.

**See also**

Analysis Explorer window, page 38

Shapefile Editor window, page 41

Samples Editor window, page 42

## 3.1.3 Semantic Net Window

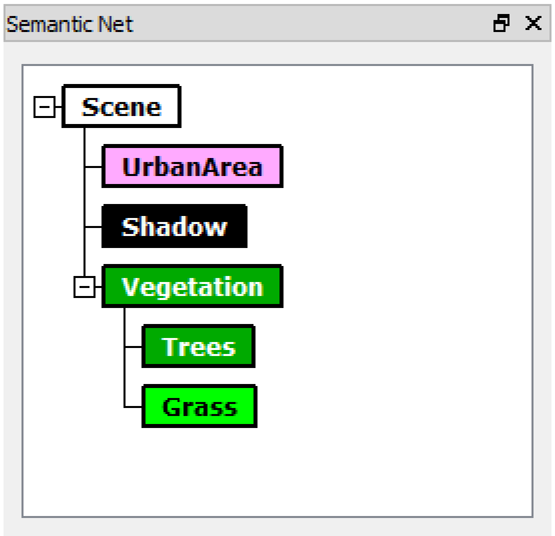


Figure 3.7 – **Semantic Net** window.

This window (Figure 3.7) allows you to interactively create and edit a semantic network.

When clicking on a node with the right mouse button, a context menu (Figure 3.8) is displayed with the following options:

**Copy** – Copies the semantic node.

**Cut** – Cuts the semantic node.

**Paste** – Pastes a node in the position of the selected node.

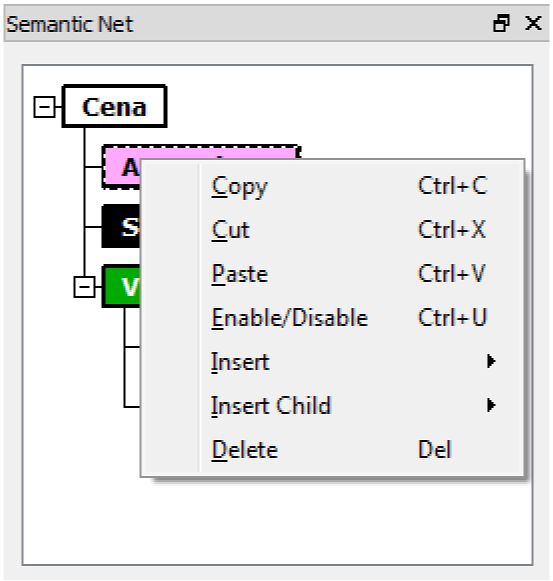


Figure 3.8 – Context menu.

**Enable/Disable** – Enables/disables the semantic node. When disabled, the semantic node is ignored during the interpretation.

**Insert** – Inserts a node in the position of the selected node.

**Insert Child** – Inserts a child-node in the selected node.

**Delete** – Removes the selected node.

**Tip**

Dragging and dropping node **A** over node **B** with the:

**Left button** – moves node **A** to node **B**'s position.

**Right button** – makes **A** a child node of **B**.

## 3.1.4 Layers Window

This window (Figure 3.9) allows you to edit and add layers to the viewer.

### 3.1.4.1 Image Tab

**Image** – Selects one of the project images.

**Keyname** – Defines the layer nickname.

**Composition** – Selects the image bands composition for visualization.

### 3.1.4.2 Shape Tab

**Keyname** – Defines the layer nickname.

**Color** – Selects the polygons color.

**Opacity** – Sets the polygons opacity. Minimum makes objects transparent.

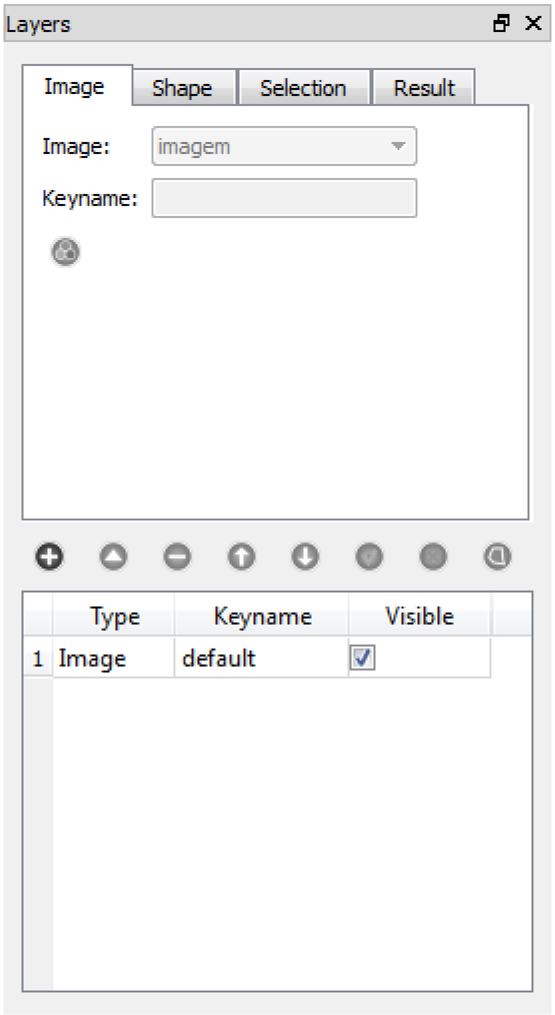


Image: imagem

Keyname:

|  | Type | Keyname | Visible |
|---|---|---|---|
| 1 | Image | default | ☑ |

Figure 3.9 – **Layers** window.

**Border** – Sets if the polygons border will be displayed or not. Allows also selecting the color of the border.

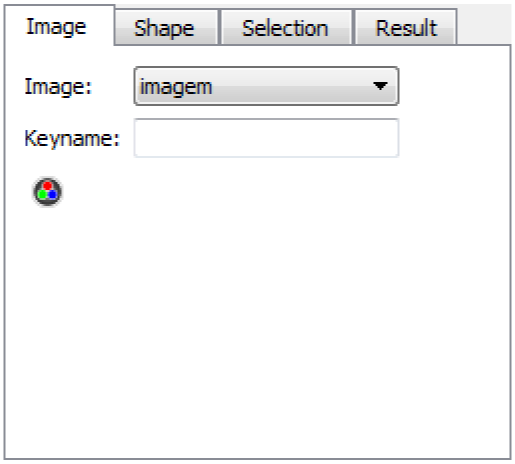
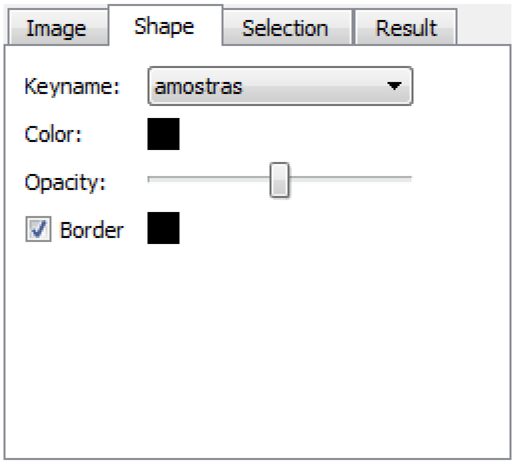


Image: imagem

Keyname:

Figure 3.10 – **Image** tab.



Keyname: amostras

Color: ■

Opacity:

☑ Border ■

Figure 3.11 – **Shape** tab.

## 3.1.4.3 Selection Tab

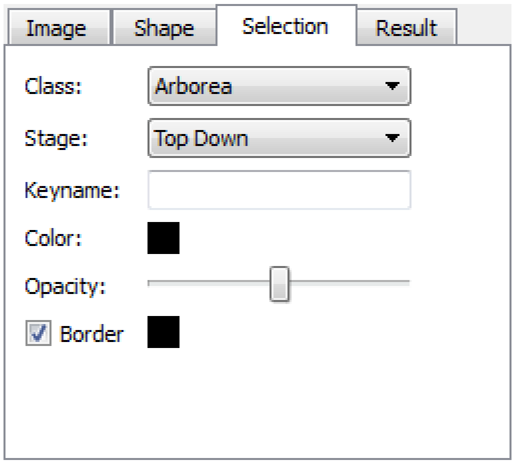
**Class** – Selects one of the semantic net classes.

**Keyname** – Defines the layer nickname.

**Color** – Selects the polygons color.

**Opacity** – Sets the polygons opacity. Minimum makes objects transparent.

**Border** – Sets if the polygons border will be displayed or not. Allows also selecting the color of the border.



Class: Arborea

Stage: Top Down

Keyname:

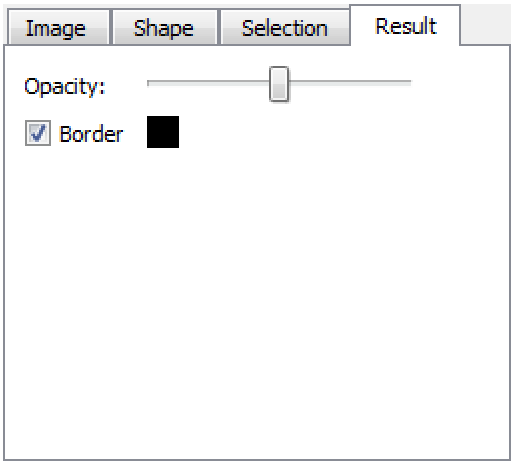
Color: ■

Opacity:

☑ Border ■

Figure 3.12 – **Selection** tab.

## 3.1.4.4 Result Tab



Opacity:

☑ Border ■

Figure 3.13 – **Result** tab.

**Opacity** – Sets the polygons opacity. Minimum makes objects transparent.

**Border** – Sets if the polygons border will be displayed or not. Allows also selecting the color of the border.

## 3.1.4.5 Toolbar

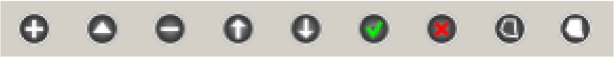


Figure 3.14 – Toolbar.

⊕ **Add** – Adds a layer to the viewer.

**Edit –** Edits the selected layer.

**Remove** – Removes the selected layer.

**Move Up** – Move the selected layer upwards.

**Move Down** – Move the selected layer downwards.

**Save** – Confirm the changes made to the selected layer.

**Cancel** – Cancel the changes made to the selected layer.

**Export as Shapefile –** Exports the selected layer as a shapefile.

**Export as Mask Image** – Exports the selected layer as a binary mask image.

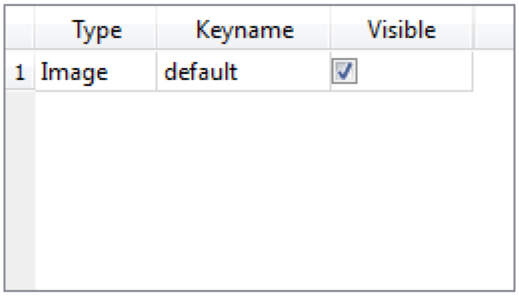
**Note**

For layers of type **Shape**, **Selection** and **Result**, the **Export as Shapefile** function allows exporting the objects to a shapefile and calculating attributes.

**See also**

Decision Rule window, page 27

### 3.1.4.6 Layers List

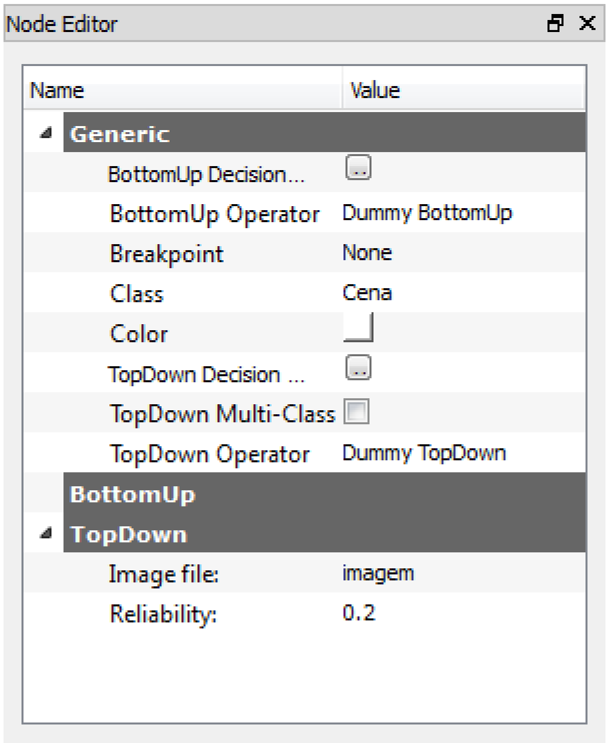


| | Type | Keyname | Visible |
|---|---|---|---|
| 1 | Image | default | ☑ |

Figure 3.15 – Layers list.

This control (Figure 3.15) displays the layers in the order they are arranged in the viewer. The **Visible** option lets you define whether the layer is visible.

## 3.1.5 Node Editor Window



Node Editor

| Name | Value |
|---|---|
| **Generic** | |
| BottomUp Decision... | ... |
| BottomUp Operator | Dummy BottomUp |
| Breakpoint | None |
| Class | Cena |
| Color | |
| TopDown Decision ... | ... |
| TopDown Multi-Class | ☐ |
| TopDown Operator | Dummy TopDown |
| **BottomUp** | |
| **TopDown** | |
| Image file: | imagem |
| Reliability: | 0.2 |

Figure 3.16 – **Node Editor** window.

This window (Figure 3.16) allows you to edit the properties of the semantic net nodes.

**BottomUp Decision Rule** – Edits the node bottom-up decision rule.

**BottomUp Operator** – Selects the node bottom-up operator.

**Breakpoint** – Defines whether the selected semantic node is an interpretation breakpoint.

**Class** – Defines the node class.

**Color** – Defines the node color.

**TopDown Decision Rule** – Edits the node top-down decision rule.

**TopDown Multi-Class** – Defines whether the top-down operator/decision rule associated to the selected node is multi-class.

**TopDown Operator** – Selects the node top-down operator.

**Note**

You may have noticed that **Node Editor** is divided into three groups of properties: **Generic**, **BottomUp** and **TopDown**. The scope of this manual is limited to the generic properties of the node. For information about operators and their parameters visit
http://wiki.dpi.inpe.br/doku.php?id=interimage:operators_documentation
http://wiki.dpi.inpe.br/doku.php?id=interimage:brec_operators_documentation.

**See also**

Decision Rule window, page 27

### 3.1.6 Viewer

This window (Figure 3.17) allows viewing and interacting with the layers added to the viewer.

**Information** – Shows the values of the selected image pixel. If an object is selected, shows its ID.

**Geocoordinates** - Shows the geographic coordinates while moving the mouse over the viewer.

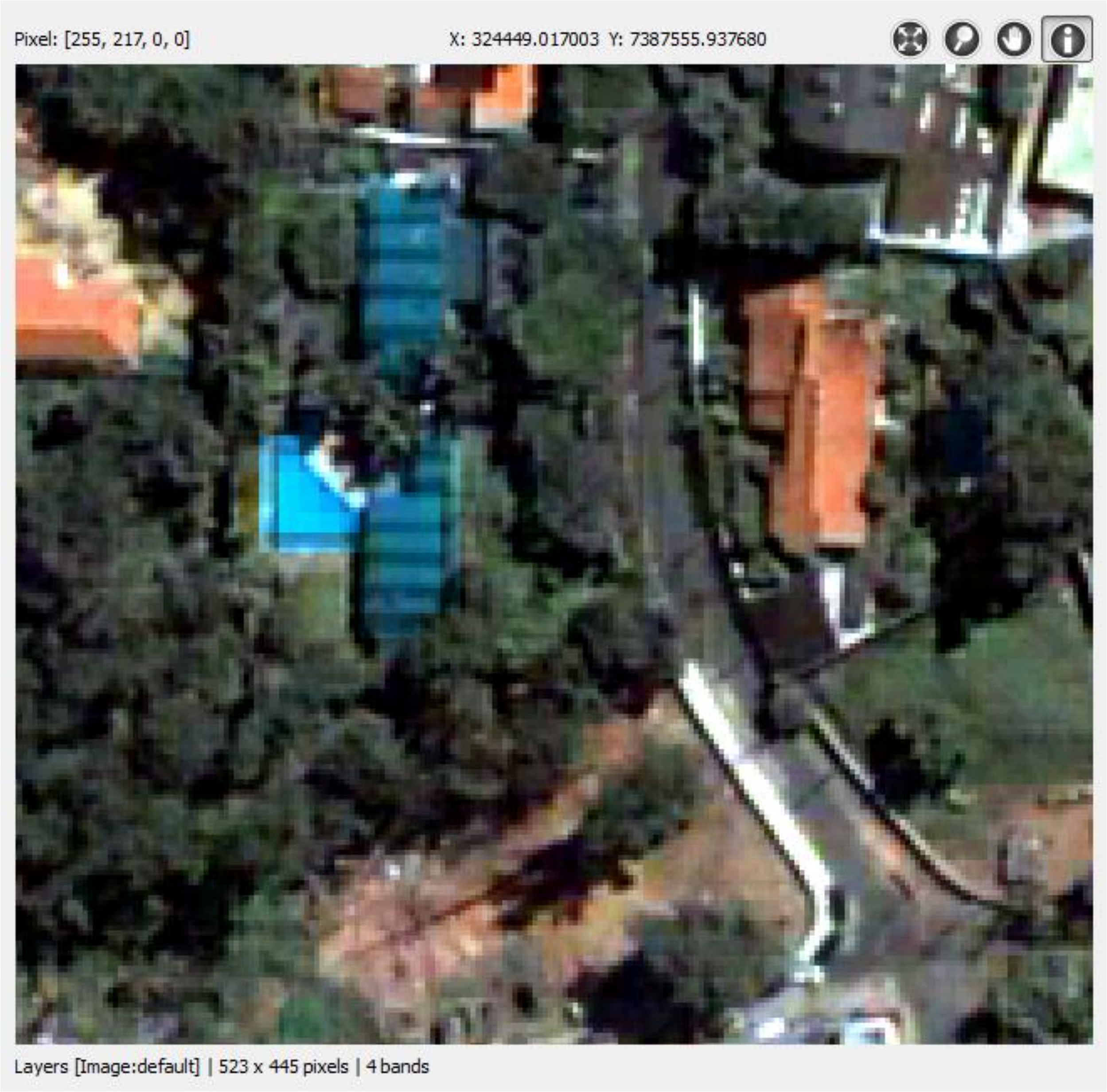
**Fit to Window** - Centers the image in the viewer.

**Zoom** - Zooms in by clicking the left mouse button. Zooms out by clicking the right one. Allows focusing on a specific part of the image by selecting it with the left button.

**Pan** - Moves the image by dragging the mouse.

**Information** – Allows showing information about the selected image or object.

**Layer Information** – If an image layer is selected, shows information about the image. Otherwise, shows information about the layer objects.



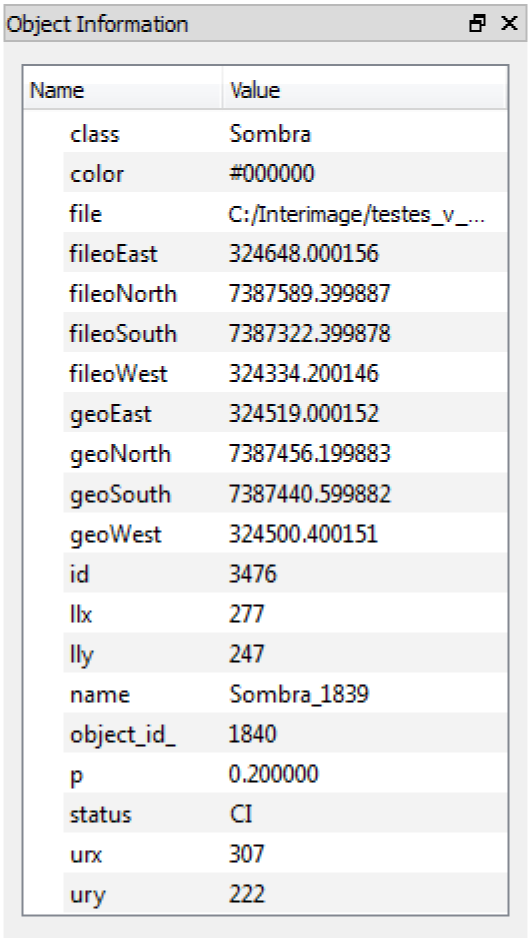
Pixel: [255, 217, 0, 0]

X: 324449.017003 Y: 7387555.937680

Layers [Image:default] | 523 x 445 pixels | 4 bands

Figure 3.17 – Viewer.

### 3.1.7 Object Information Window

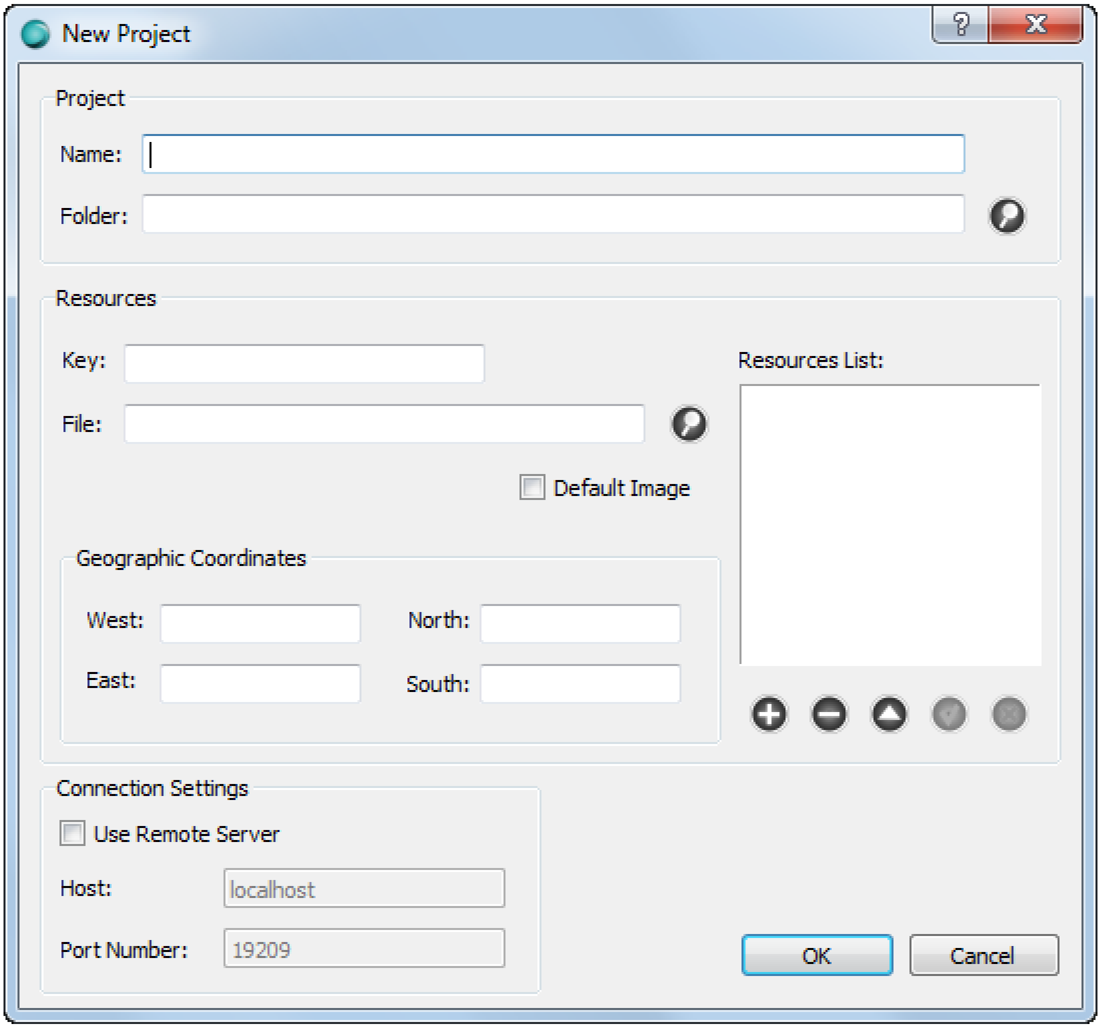


| Name | Value |
|---|---|
| class | Sombra |
| color | #000000 |
| file | C:/Interimage/testes_v_... |
| fileoEast | 324648.000156 |
| fileoNorth | 7387589.399887 |
| fileoSouth | 7387322.399878 |
| fileoWest | 324334.200146 |
| geoEast | 324519.000152 |
| geoNorth | 7387456.199883 |
| geoSouth | 7387440.599882 |
| geoWest | 324500.400151 |
| id | 3476 |
| llx | 277 |
| lly | 247 |
| name | Sombra_1839 |
| object_id_ | 1840 |
| p | 0.200000 |
| status | CI |
| urx | 307 |
| ury | 222 |

Figure 3.18 – **Object Information** window.

This window (Figure 3.18) allows you to view the properties of the selected object in the viewer.

## 3.2 New/Edit Project Window



Host: localhost

Port Number: 19209

Figure 3.19 – **New Project** window.

This window (Figure 3.19) allows you to create or edit an interpretation project. It offers the following options:

**Name** – Defines the project name.

**Folder** – Defines the project folder.

**Key** – Defines a resource nickname.

**File** – defines the resource file.

**Default Image** - Although allowing multiple images, InterImage requires that one of them is selected as default image.

**West, North, East, South** - If the resource is an image, defines its geographical coordinates.

**Resources List –** Lists the project resources.

⊕ **Add –** Adds a resource to the project.

⊖ **Remove –** Removes the selected resource from the project.

▲ **Edit –** Edits the selected resource.

✓ **Save –** Confirms the changes made to the selected resource.

✗ **Cancel** - Cancels the change made to the selected resource.

**Use Remote Server –** Enables the project to communicate with a server.

**Host –** Defines the host address.

**Port Number –** Defines the communication port.

## 3.2.1 Supported Resource Formats

Table 3.1 presents the resources formats that can be used in a project:

**Note**

Give preference to image formats that do not have compression (**. tif**) because they offer a higher quality than **.jpg**, which is compressed. In this way, you will get better results.

| **Format** | **Extension** |
|---|---|
| Tagged Image File | **.tif** |
| Tagged Image File (Geo Tiff) | **.tif** |
| ESRI Shape File | **.shp** |
| ESRI Shape File | **.dbf** |
| JPEG JFIF | **.jpg** |

Table 3.1 – Supported resource formats.

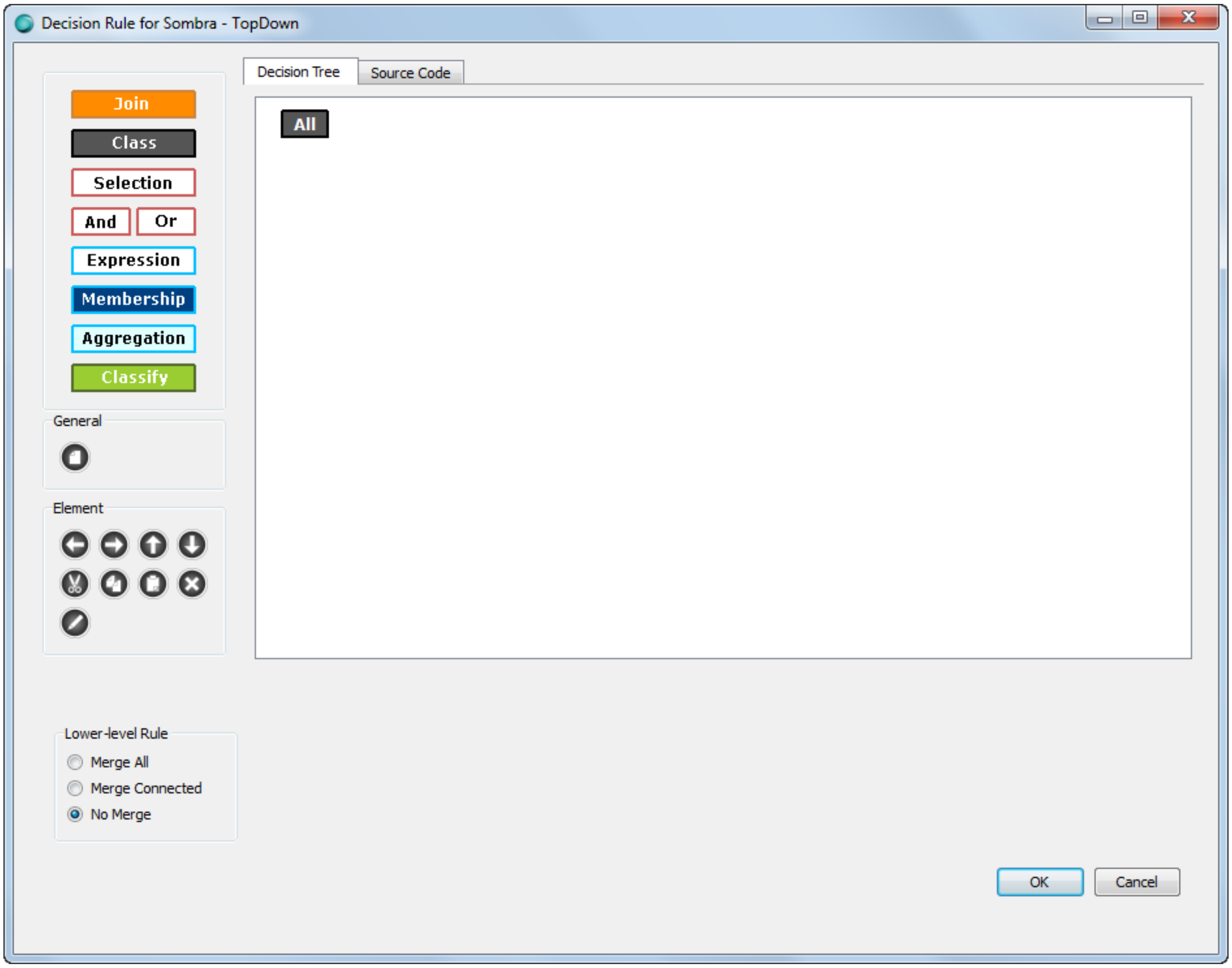
## 3.2.2 Maximum Image Size

It is hard to set an upper limit since the answer to this question involves other quantities such as the size of the semantic net or the number of objects produced by the interpretation. In our tests, however, we could successfully interpret images from up to 9 Megapixels (3,000 x 3,000 pixels).

## 3.2.3 Resources in Different Resolutions

InterImage is able to work with shapefiles and images of different sizes, resolutions and geographic coverage. In the case of images, those of lower resolution - larger pixel size - are resampled according to the higher resolution.

## 3.3 Decision Rule Window



Lower-level Rule

○ Merge All
○ Merge Connected
● No Merge

Figure 3.20 – **Decision Rule** window.

This window (Figure 3.20) allows you to create a set of expressions, called decision rule. These expressions define the structured and explicit knowledge of the user/analyst and are used by the system in the interpretation process.

### 3.3.1 Building Blocks

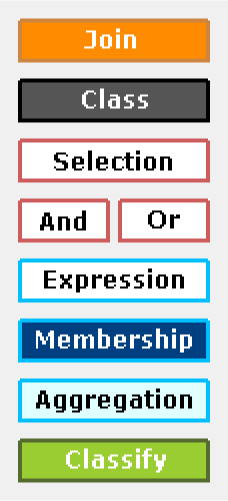


Figure 3.21 – Building blocks.

The decision rule is constructed through a set of building blocks (Figure 3.21):

**Join** – Joins several **Class** blocks.

**Class** – Select objects of a particular class or classes.

**Selection** – Selects objects that meet a certain criterion.

**And** and **Or** – Allow you to create powerful logical expressions. They are used along with the **Selection** block.

**Expression** – Allows creating a new attribute from the result of a mathematical expression.

**Membership** – Works with fuzzy logic.

**Aggregation** – Allows a level of the semantic net to pass information to the upper level.

**Classify** - Classifies objects and solve spatial conflicts. It is usually the last block of the rule.

**Tip**

To add a block, just click on it and it will be inserted at the end of the rule. If the block is not inserted, it might not be allowed in this position. So you need to drag the block and drop it over other block in the rule respecting this convention:

**Left Button** – moves block **A** to block **B** position.

**Right Button** – turns block **A** son of block **B**.

## 3.3.2 Toolbar

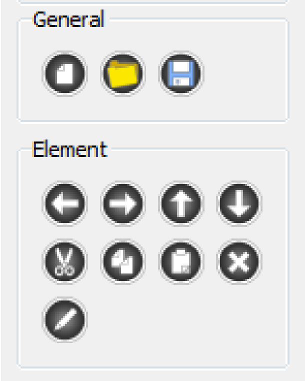


Figure 3.22 – Toolbar.

**New** – Erases the current rule and initializes a new one.

**Open Decision Rule** – Opens a decision rule.

**Save Decision Rule** – Saves the current decision rule.

**Level Up** – Moves the block one level backwards.

**Level Down** – Moves the block one level forward.

**Move Up** - Moves the selected block upwards within the same tree level.

**Move Down** – Moves the selected block downwards within the same tree level.

**Cut** – Cuts the selected block.

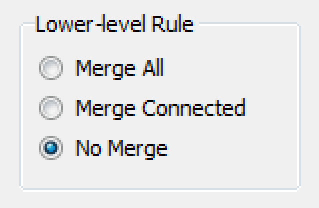
**Copy** – Copies the selected block.

**Paste** – Pastes a block in the position of the selected block.

**Delete** – Removes the selected block.

**Edit** – Edits the selected block.

## 3.3.3 Upper/Lower Level Rule



Lower-level Rule

○ Merge All

○ Merge Connected

● No Merge

Figure 3.23 – Lower-level rule.

This control (Figure 3.23) defines how objects will be received by the higher/lower level of the net.

**Merge All** - Join all objects of the same class, though disjointed, into a single larger object.

**Merge Connected –** Joins connected objects of the same class in larger objects.

**No Merge** - No object is changed.

## 3.3.4 Decision Tree Tab

This tab (Figure 3.24) provides a powerful interface to interactively build and edit the decision tree.

By clicking on a block with the right mouse button, a context menu is displayed (Figure 3.25) with the following options:

**Edit –** Edits the selected block.

**Copy –** Copies the selected block.

**Cut –** Cuts the selected block.

**Paste –** Pastes a block in the position of the selected block.

**Comment/Uncomment –** Comments the selected block (and its children) so it’s ignored in the rule execution.

**Insert –** Inserts a block in the position of the selected block.

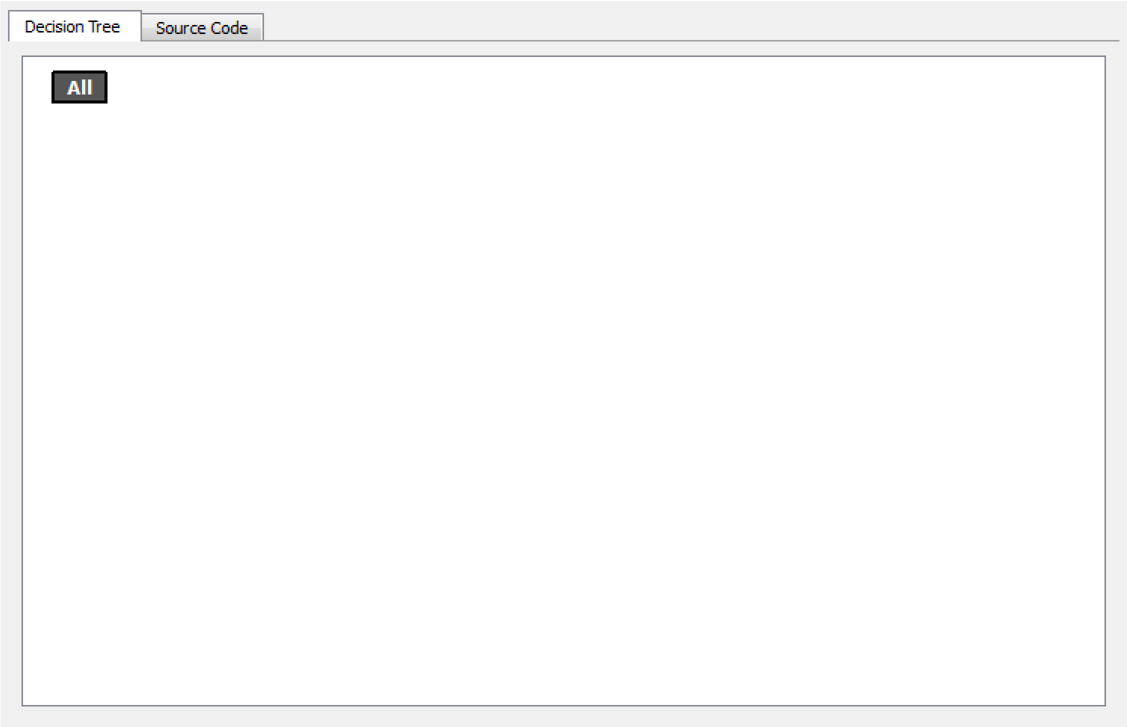


Figure 3.24 – **Decision Tree** tab.

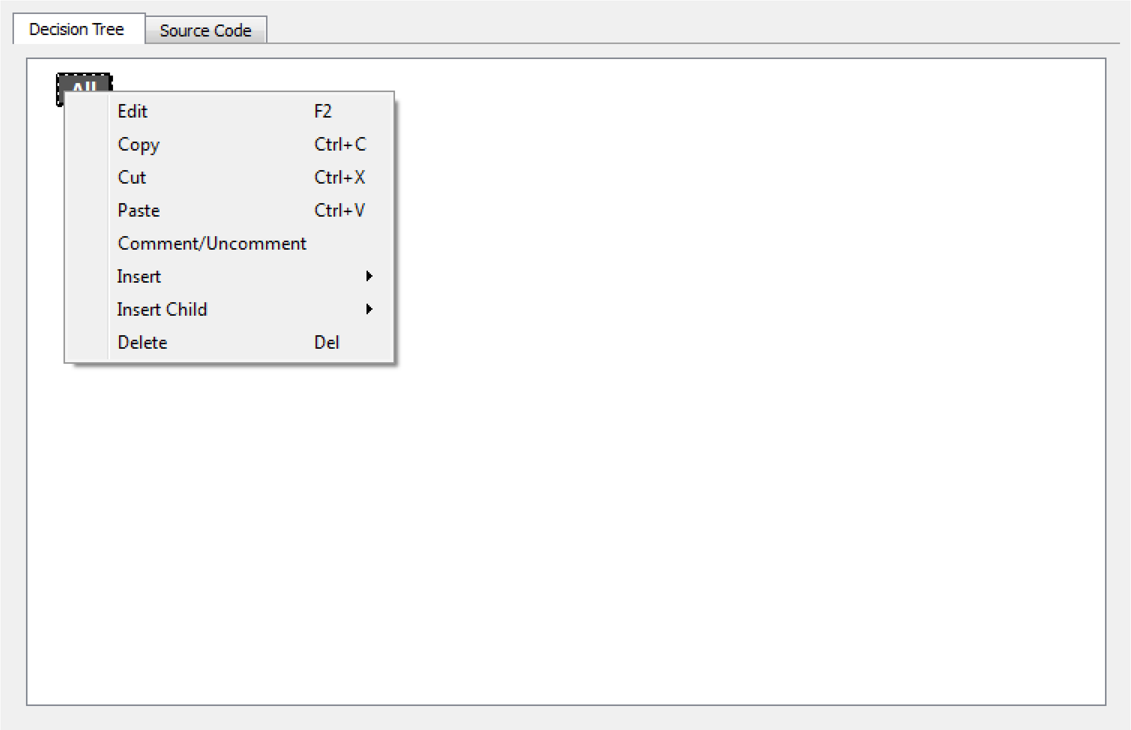


Figure 3.25 – Context menu.

**Insert Child** – Inserts a child-block in the selected block.

**Delete** – Removes the selected block.

**Tip**

Dragging and dropping block **A** over block **B** with the:

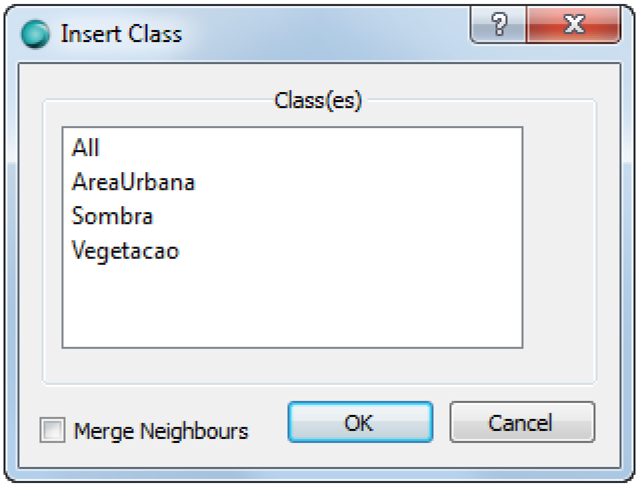
**Left Button** – moves block **A** to block **B** position.

**Right Button** – turns block **A** son of block **B**.

## 3.3.5 Source Code Tab

This tab was used in earlier versions to allow advanced users to edit the decision rule directly in its original form in Reverse Polish Notation. In this version, this is no longer possible and, probably, this tab will be removed in future versions.

## 3.3.6 Insert/Edit Class Window



All
AreaUrbana
Sombra
Vegetacao

□ Merge Neighbours

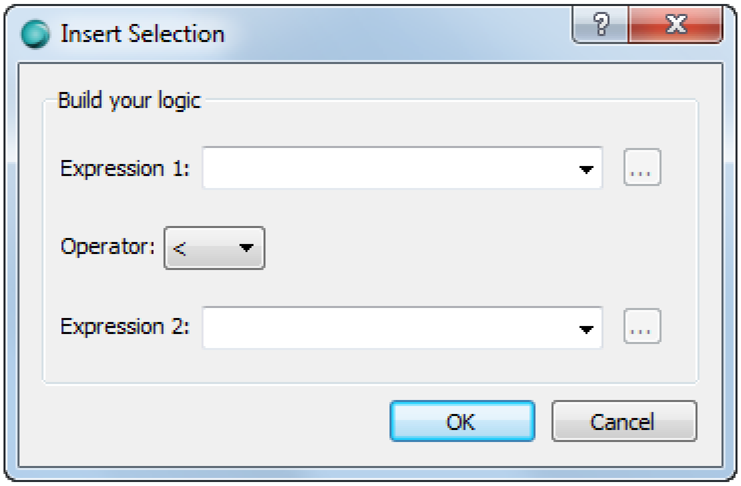
Figure 3.26 – **Insert Class** window.

This window (Figure 3.26) allows you to select the objects of a particular class or classes.

**Class(es)** - Select one or more (using the *Ctrl* key) classes.

**Merge Neighbors** – Groups connected objects into larger objects.

## 3.3.7 Insert/Edit Selection Window



Operator: <

Figure 3.27 – **Insert Selection** window.

| Expression 1 | Operator | Expression 2 |
|---|---|---|
| Permite +, -, ×, ÷, (, ) | <, >, ≤, ≥, =, ≠ | Permite +, -, ×, ÷, (, ) |

Table 3.2 – Selection criterion.

This window (Figure 3.27) allows you to select objects that meet a certain criterion. This criterion is of the form:

**Expression 1** – Defines an attribute or expression.

**Operation** – Defines a logical operator.

**Expression 2** – Defines another attribute or expression.

## 3.3.8 Insert/Edit Expression Window

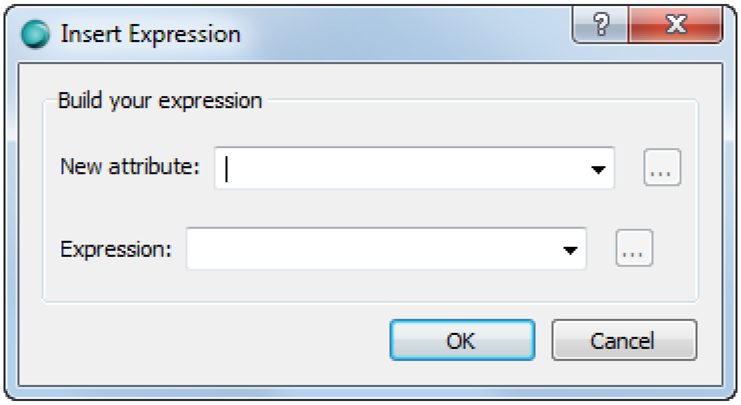


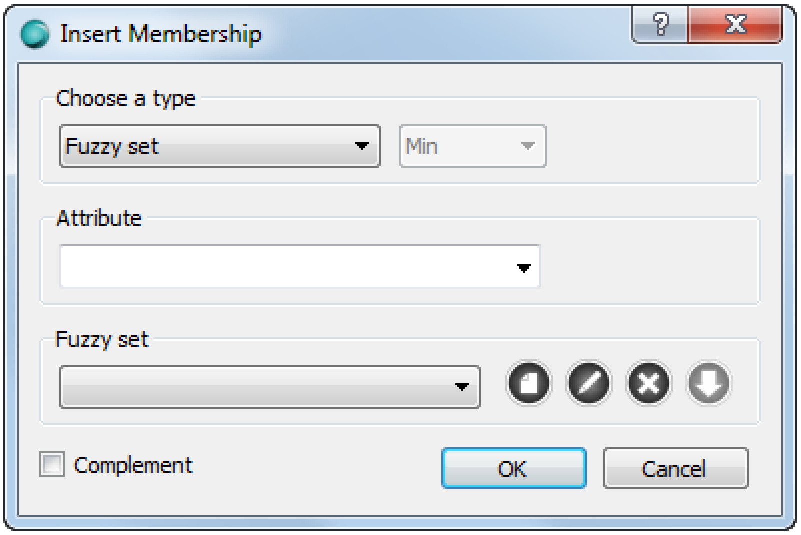
Figure 3.28 – **Insert Expression** window.

This window (Figure 3.28) allows you to create a new attribute from another attribute or from the result of a mathematical expression.

**New attribute** – Defines the new attribute name.

**Expression** – Defines an attribute or expression.

## 3.3.9 Insert/Edit Membership Window



Choose a type: Fuzzy set, Min

Figure 3.29 – **Insert Membership** window.

This window (Figure 3.29) allows you to create sets and expressions of fuzzy logic.

**Type** - Defines the type of **Membership** block. It has the following options:

**Fuzzy set** - Inserts a block that returns the membership value of an attribute to the selected fuzzy set.

**Operation** – Inserts a fuzzy operation: **Min**, **Max**, **Mean**, **Mul**, **Sum**.

**Membership value** - Inserts a block with a membership value defined by the user.

**Operator** – Defines the fuzzy operator.

**Attribute** – Defines the fuzzy set input attribute.

**Fuzzy set** – Defines the fuzzy set.

**New** – Creates a new fuzzy set.

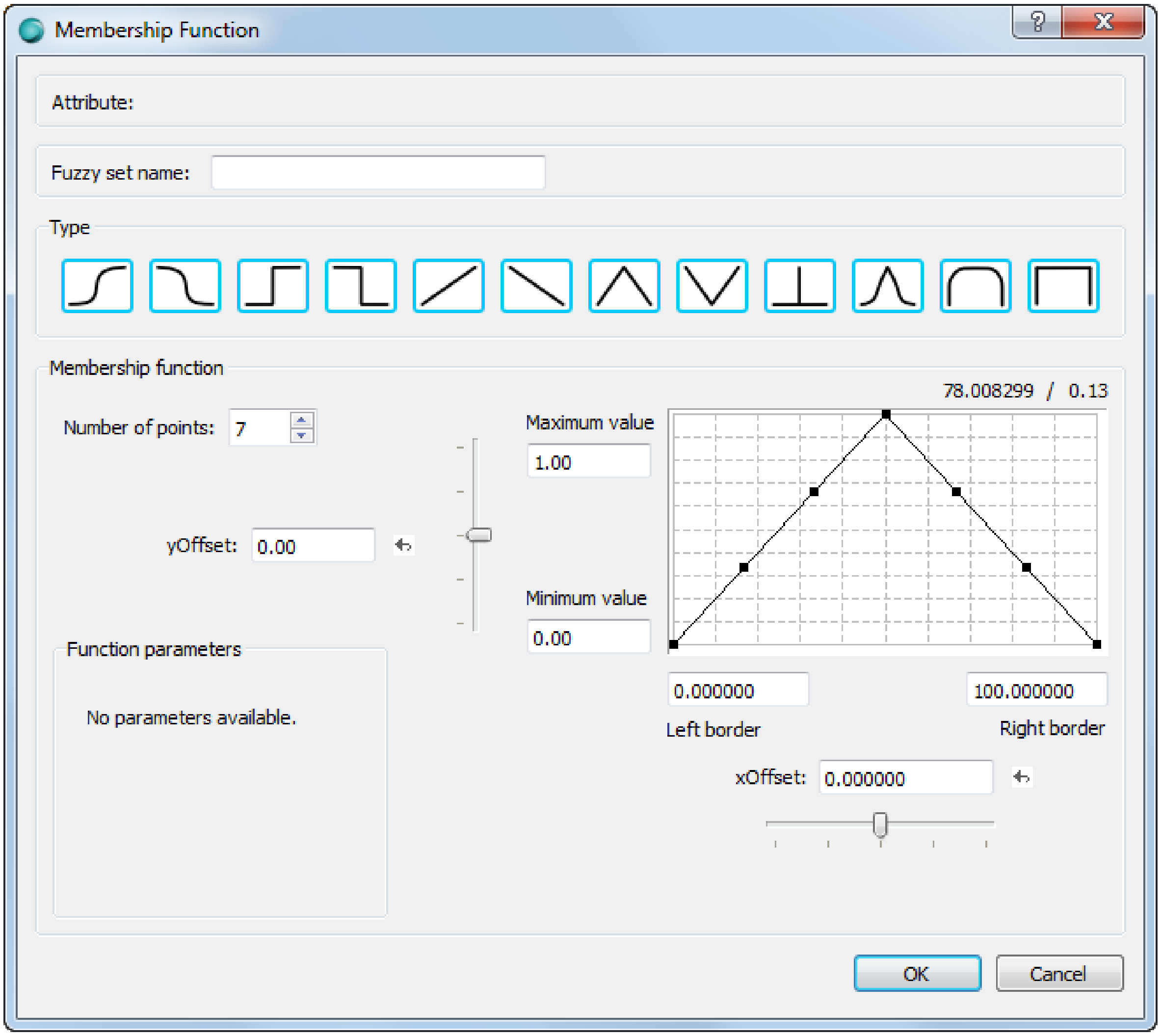
**Edit** – Edits the selected fuzzy set.

**Delete** – Removes the selected fuzzy set.

**Import** – Allows importing a fuzzy set from a **.fuzz** file (Not implemented).

**Complement** - Returns the complement of the computed membership value.

## 3.3.10 Membership Function Window



Attribute:

Fuzzy set name:

Number of points: 7

yOffset: 0.00

No parameters available.

78.008299 / 0.13

Maximum value: 1.00

Minimum value: 0.00

Left border: 0.000000

Right border: 100.000000

xOffset: 0.000000

Figure 3.30 – **Membership Function** window.

This window (Figure 3.30) allows you to create and edit the membership function of a fuzzy set.

**Attribute –** Shows the fuzzy set input attribute.

**Fuzzy set –** Defines the fuzzy set name.

**Type** - Defines the shape of the membership function (Table 3.3).

**Number of points** - Defines the number of points used to draw the function: 9, 11, 13 or 15.

**yOffset** - Defines the y-axis offset.

**Maximum value –** Maximum membership value.

**Minimum value –** Minimum membership value.

**Membership Function** - Allows you to edit the membership function. Just drag the vertices with the mouse.

**Left border** - Lower limit of the function domain.

**Right border –** Upper limit of the function domain.

**xOffset –** Defines the x-axis offset.

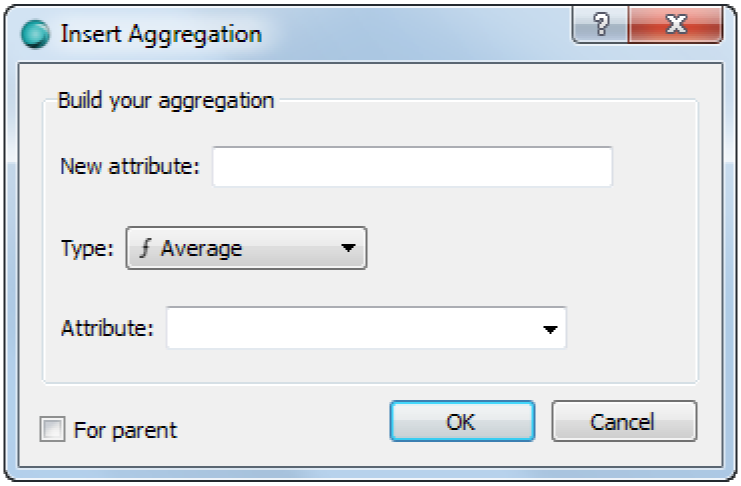
**Function parameters** - Some functions allow a fine adjustment of parameters such as **slope**, **inflexion point**, **mean** e **standard deviation**.

| Button | Form |
|---|---|
| ⎱ | Greater than |
| ⎰ | Lower than |
| _\|‾ | Greater than (crisp) |
| ‾\|_ | Lower than (crisp) |
| ╱ | Greater than (linear) |
| ╲ | Lower than (linear) |
| ∧ | Linear range (triangle) |
| ∨ | Linear range (inverted triangle) |
| ⊥ | Singleton (exact value) |

| | |
|---|---|
| ∧ | Approximate gaussian |
| ∩ | Approximate range |
| ⊓ | Complete range |

Table 3.3 – Standard forms of membership functions.

## 3.3.11 Insert/Edit Aggregation Window


Insert Aggregation

Build your aggregation

New attribute:

Type: ʃ Average

Attribute:

☐ For parent

Figure 3.31 – **Insert Aggregation** window.

This window (Figure 3.31) allows you to create a new attribute from another attribute aggregation. The new attribute can also be passed to the upper level of the net.

**New attribute** – Defines the new attribute name.

**Type** – Defines the aggregation type: **Average**, **Standard deviation**, **Maximum**, **Sum**, **Division**, **Count**.

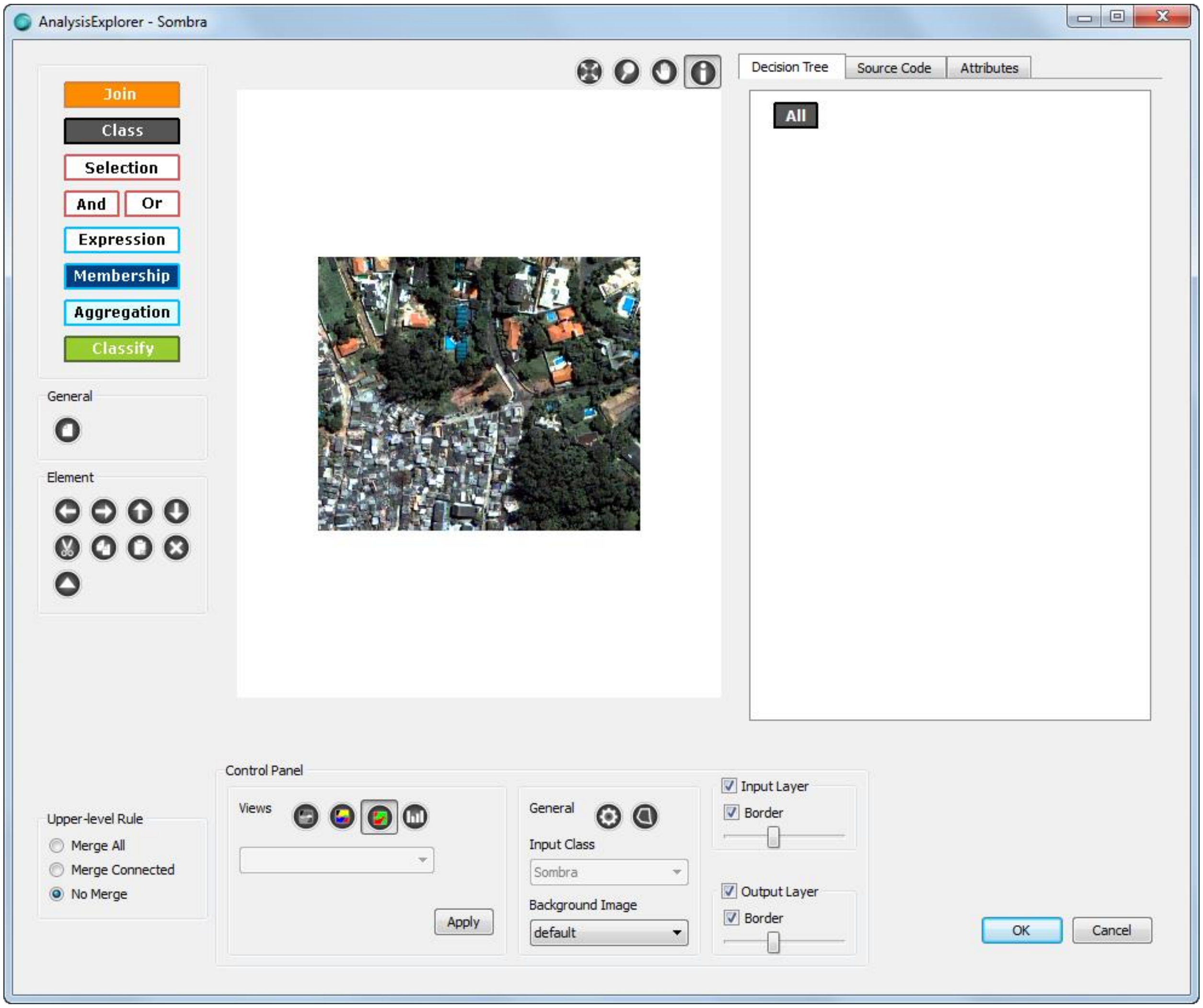
**Attribute** – Defines the attribute to be aggregated.

**For parent** – Defines if the new attribute will be passed to the upper level of the net.

**Note**

For information on the attributes (features) that you can compute using InterImage visit http://wiki.dpi.inpe.br/doku.php?id=interimage:attributes_description.

## 3.4 Analysis Explorer Window



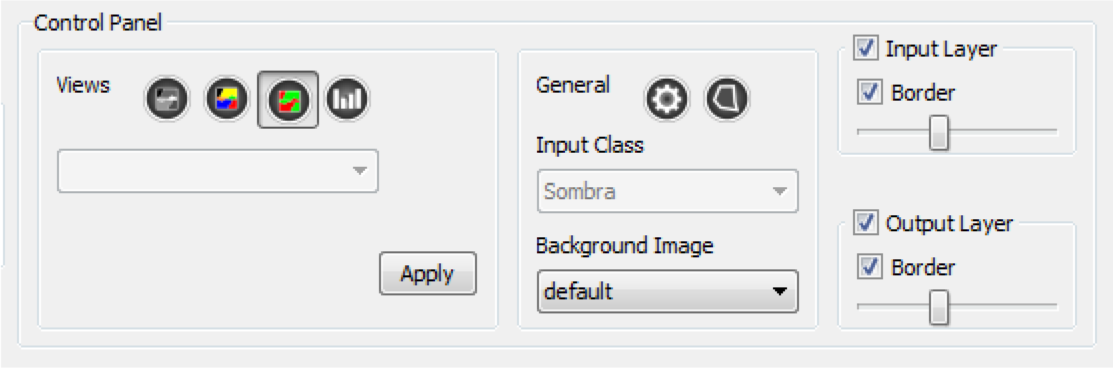
Upper-level Rule
○ Merge All
○ Merge Connected
● No Merge

Input Class: Sombra
Background Image: default
☑ Input Layer
☑ Border
☑ Output Layer
☑ Border

Figure 3.32 – **Analysis Explorer** window.

This window (Figure 3.32) offers tools that help in building the interpretation model. It is a variation of the **Decision Rule** window (Section 3.3), therefore it has an interface that allows creating a decision rule. However, it adds a **Control Panel** that has tools that allow us to analyze the characteristics of objects and the result of the rule and help, thus, building the model. A new **Attributes** tab is also added in the upper right control. It allows you to inspect the properties of the selected node in the viewer.

## 3.4.1 Control Panel



Input Class: Sombra

Background Image: default

☑ Input Layer

☑ Border

☑ Output Layer

☑ Border

Figure 3.33 – **Control Panel**.

**Views** – Switches display modes:

**Attribute View** – Spatializes in grayscale the attribute selected in the **Attribute** field. Being the object that has the lowest value of the attribute in black and the highest value in white.

**Classification View** - Displays objects with the colors of the respective classes that have been associated to them.

**Selection View** - Shows in red all objects that existed before the rule execution (input), and in green those that remained after that (output).

**Analyis Tools** - Opens a window that allows analyzing the statistical distribution of the segments attributes and thus making better decisions about how to build the rule.

**Attribute** - Selects the attribute to be used in **Attribute View** mode.

**Apply** - Applies the selected exhibition mode to the viewer.

**Execute** – Executes the decision rule.

**Export** - Exports the result of the rule to a shapefile.

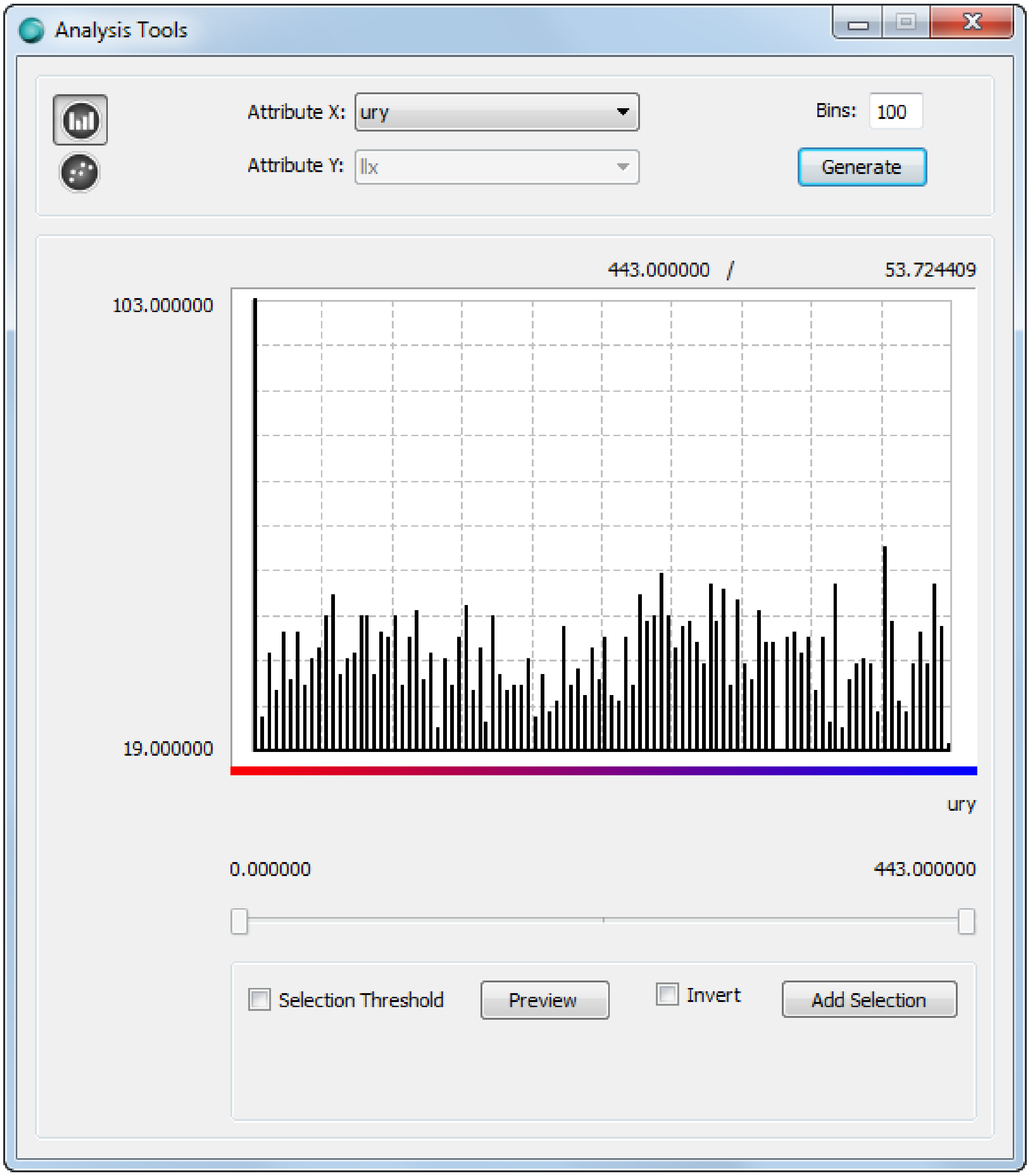
**Input Class** - When there is more than one class in the rule input, this control allows you to select which class will be displayed in the viewer.

**Background Image** - Selects the image that will be displayed in the viewer.

**Input Layer** - Enables/disables the display of the input layer. The **Border** field allows you to enable/disable viewing the objects border and the third control sets the polygons opacity. Minimum value makes objects transparent.

**Output Layer** - Enables/disables the display of the output layer. The **Border** field allows you to enable/disable viewing the objects border and the third control sets the polygons opacity. Minimum value makes objects transparent.

## 3.4.2 Analysis Tools Window



Attribute X: ury

Attribute Y: llx

Bins: 100

☐ Selection Threshold

☐ Invert

Figure 3.34 – **Analysis Tools** window.

This window provides tools for analyzing the statistical distribution of the segments attributes.

**Histogram** - Selects the histogram display mode.

**Scatter Plot** – Selects the scatter plot display mode.

**Attribute X** - Selects the attribute which histogram will be displayed. In **Scatter Plot** mode, selects the attribute of the x-axis which will be combined with **Attribute Y**.

**Attribute Y** – Selects the y-axis attribute to generate the scatter plot.

**Bins** – Defines the number of histogram bars.

**Generate** – Generates the graph.

**Selection Threshold/Line** – Allows setting a point in histogram mode or a line in scatter plot mode that performs a segments selection. Those which stay in the red side are excluded and those which stay in the blue side remain.

**Preview** - Shows in the **Analysis Explorer** window the result of the selection made with the chosen threshold.

**Invert** – Inverts the objects selection criterion.

**Add Selection** – Creates a **Selection** block in the decision rule with the generated selection expression.

## 3.5 Shapefile Editor Window

This window (Figure 3.35) allows creating and editing *shapefiles*.

**Load ESRI Shapefile** - Loads a *shapefile*.

**Save ESRI Shapefile** – Saves the polygons to a *shapefile*.

**Save Mask Image** - Saves the polygons as a binary mask in PBM format. This mask can be used in **Samples Editor** to define a region of interest.

**Clear All** – Removes all polygons.

**Zoom** - Zooms in by clicking the left mouse button. Zooms out by clicking the right one. Allows focusing on a specific part of the image by selecting it with the left button.

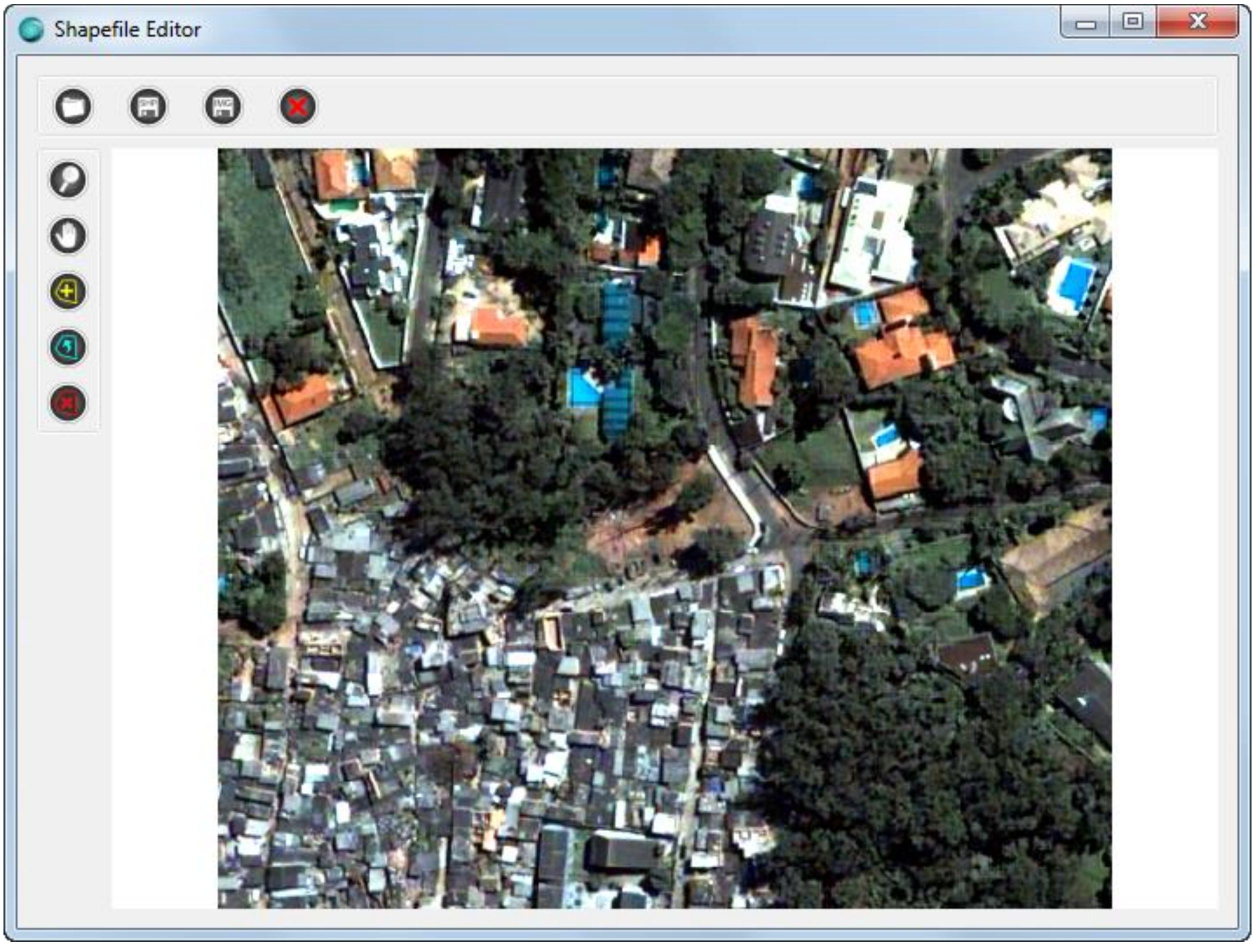


Figure 3.35 – **Shapefile Editor** window.

**Pan** - Moves the image by dragging the mouse.

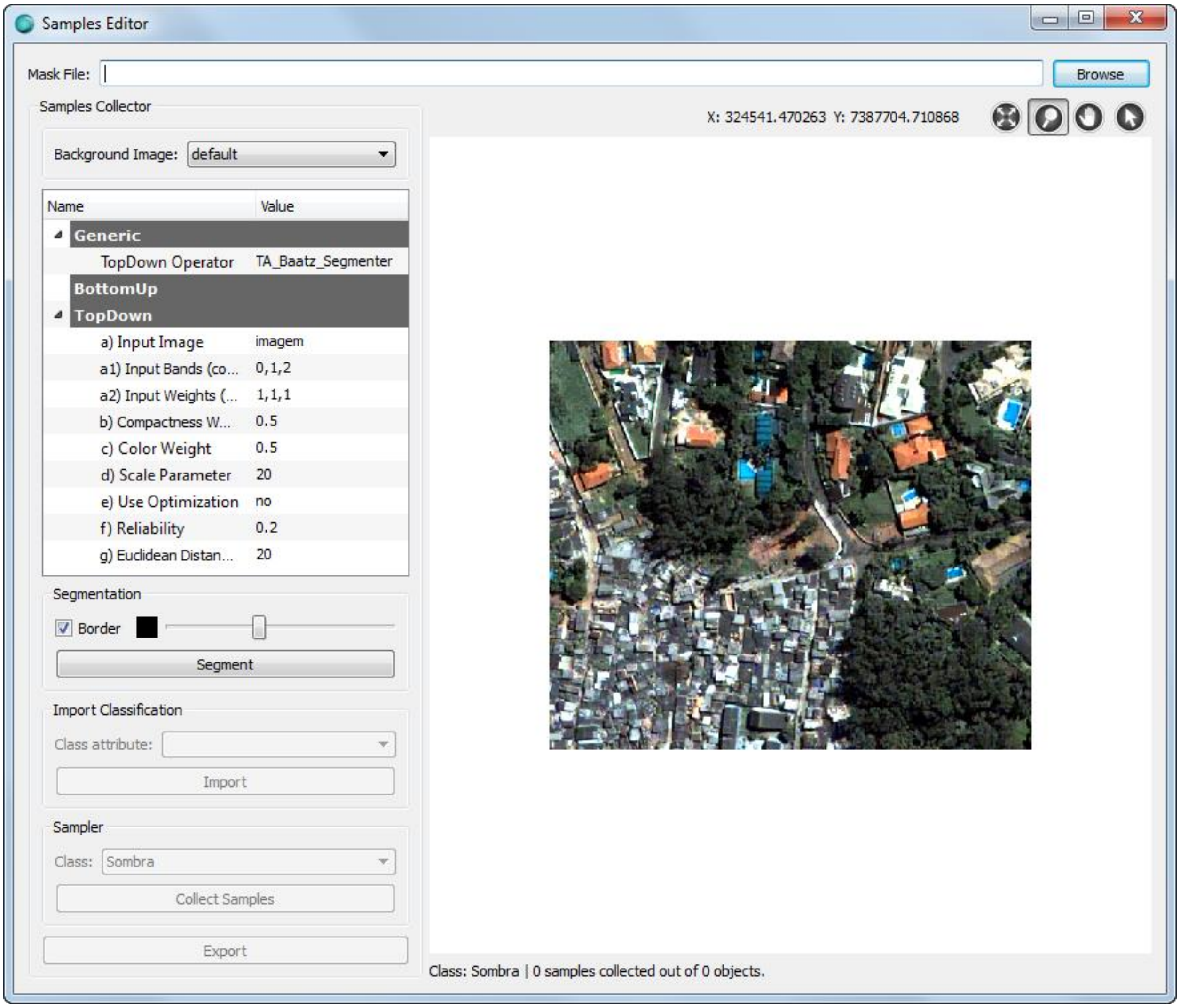
**Create Polygon** - Enables the polygons creation mode. To create a polygon, click the left mouse button to create the vertices. To close the polygon, double click or click on the starting point. And to undo a vertex, click the right button.

**Edit Polygon** - Enables the polygons editing mode. This mode does not allow you to add or remove points, just to move them.

**Delete Polygon** - Enters the polygons removal mode. To remove a polygon, just click on it.

## 3.6 Samples Editor Window

This window (Figure 3.36) allows performing a segmentation, collecting samples and classifying them manually, generating a shapefile at the end of the process.



Mask File:

X: 324541.470263 Y: 7387704.710868

Background Image: default

| Name | Value |
|---|---|
| Generic | |
| TopDown Operator | TA_Baatz_Segmenter |
| BottomUp | |
| TopDown | |
| a) Input Image | imagem |
| a1) Input Bands (co... | 0,1,2 |
| a2) Input Weights (... | 1,1,1 |
| b) Compactness W... | 0.5 |
| c) Color Weight | 0.5 |
| d) Scale Parameter | 20 |
| e) Use Optimization | no |
| f) Reliability | 0.2 |
| g) Euclidean Distan... | 20 |

☑ Border

Class attribute:

Class: Sombra

Class: Sombra | 0 samples collected out of 0 objects.

Figure 3.36 – **Samples Editor** window.

**Mask File** – Allows selecting a binary mask file in **PBM** format. The segmentation will be performed within the area defined by the mask.

**Background Image** - Allows selecting which image will be used. The images available here are those defined as project resources.

**Segmentation** – Allows configuring the segmentation process.

**Segmenter** - Allows configuring the parameters for the segmentation process (see note below).

**Opacity** - Sets the polygons opacity. Minimum makes objects transparent.

**Border** – Sets if the polygons border will be displayed or not. Allows also selecting the color of the border.

**Segment** – Executes the segmentation process.

**Import Classification** – Allows importing the classification information present in the polygons. This functionality should be used only after running the **Import Samples** operator.

**Class Attribute** – Attribute that contains the class information. It is only necessary when the attribute name is other than **class**, otherwise the **Import Samples** operator automatically recognizes the classes.

**Import** – Imports the classification information.

**Sampler** – Offers the samples collection functionality.

**Class** - Selects the class for which you want to collect the samples. The available classes are the classes present in the semantic net.

**Collect Samples** - Click this button to start collecting the samples for the selected class. To select a polygon, click on it. To deselect, click again.

**Export** - Exports the segmentation to a shapefile and allows calculating attributes.

**Viewer** – Allows visualizing and interacting with the image and the sample polygons.

**Information** – Shows the values of the selected image pixel.

**Geocoordinates** - Shows the geographic coordinates while moving the mouse over the viewer.

**Fit to Window** – Centers the image in the viewer.

**Zoom** – Zooms in by clicking the left mouse button. Zooms out by clicking the right one. Allows focusing on a specific part of the image by selecting it with the left button.

**Pan** – Moves the image by dragging the mouse.

**Selection** – Selects/deselects sample polygons.

**Samples Information** – Shows information about the samples collection.

**Note**

For information about the operators and their parameters, please visit
http://wiki.dpi.inpe.br/doku.php?id=interimage:operators documentation
http://wiki.dpi.inpe.br/doku.php?id=interimage:brec_operators_documentation.

# 4 Batch Processing

In this version it is possible to run InterIMAGE in batch mode. With this new operation mode it is possible to create an interpretation project and then run it several times without having to interact with the user interface. This was mainly done to allow the execution of the so called pipelines. So, for example, now it is possible to run a classification project and a shape regularization of the building polygons found in the classification by writing only two command lines.

This is also interesting to test a classification project on different images or using different ROIs without having to change them using the interface. In fact, it is possible to write a batch file that executes different InterIMAGE instances to classify hundreds of images with little user interaction.

Finally, using some options it is possible to run slightly different configurations of the same operator or decision rule (this could be used to tune automatically the parameters values).

**Note**

For information about the batch mode, please visit
http://wiki.dpi.inpe.br/doku.php?id=interimage:batch_processing.

# References

[1] Câmara, G., Souza, R.C.M., Pedrosa, B.M., Vinhas, L., Monteiro, A.M.V., Paiva, J.A., Carvalho, M.T., Gatass, M., 2000. TerraLib: Technology in Support of GIS Innovation. In: II Brazilian Symposium on GeoInformatics, GEOINFO 2000. São Paulo, Brazil. Proceedings of GEOINFO 2000 (CD-ROM).

[2] Gamba, P., Dell'Acqua, F., Lisini, G., 2009. BREC: The Built-up area RECognition tool. In: Joint Urban Remote Sensing Event, 2009. Page(s): 1-5.

[3] Bückner, J., Pahl, M., Stahlhut, O., Liedtke, C.-E., 2001. GEOAIDA – A knowledge-based automatic image data analyzer for remote sensing data. In: ICSC Congress on Computational Intelligence Methods and Applications 2001 - CIMA 2001, Bangor, Wales, UK. Proceedings of the Congress on Computational Intelligence Methods and Applications 2001 – CIMA 2001 (CD-ROM).

[4] Pahl, M., 2008. Arquitetura de um sistema baseado em conhecimento para a interpretação de dados de sensoriamento remoto de múltiplos sensores. PhD Thesis, University of Hannover, (Translation), INPE, São José dos Campos (INPE-15211-TAE/71) URL: http://urlib.net/sid.inpe.br/mtc-m17@80/2008/03.07.18.31 (26 Sep 2008).