eCognition® 8.0: Guided Tour – Level 3

Advanced example of building extraction

For intermediate and advanced users;
Classification of buildings using LiDAR and RGB data

Version 3.0

Legal Notices
Copyright and Trademarks

Copyright © 2010, Trimble Navigation Limited. All rights reserved. Trimble, the Globe & Triangle logo and eCognition are trademarks of Trimble Navigation Limited, registered in the United States and in other countries. All other product names, company names, and brand names mentioned in this document may be trademark properties of their respective holders.

Protected by patents US REG NO. 3,287,767; WO9741529; WO9802845; EP0858051; WO0145033; WO0205198; WO2004036337 ; EP1866849; US 6,229,920; US 7,117,131; US 6,832,002; US 7,437,004; US 7,574,053 B2; US 7,146,380; US 7,467,159 B2; US 20070122017; US 20080008349; US 12/386,380. Further patents pending.

The contents of this manual are subject to change without notice. Although every effort has been made to ensure the accuracy of this manual, we can not guarantee the accuracy of its content. If you discover items which are incorrect or unclear, please contact us using this form: http://www.ecognition.com/content/training-inquiries

Find more information about the product or other tutorials at

http://www.ecognition.com/

http://www.ecognition.com/community

Release Notice

Release date: August 2010

Table of Contents

Advanced example of building extraction 1
Table of Contents 3
Introduction 6
About this Guided Tour 6
Data Included with the Guided Tour 6
How to use this Guided Tour 7
Lesson 1 Get the big picture 10
Creating image objects 10
Classification 11
Export the result 12
Lesson 2 Loading the data 13
2.1.1 Open eCognition Developer Rule Set Mode 13
2.2 Option 1: Using workspace and import template 14
2.2.1 Create a new workspace 14
2.2.2 Import data in the workspace with an import template 15
2.3 Option 2 (Trial version): Create an individual Project 16
2.3.1 Define the data to be loaded 16
2.3.2 Define layer alias 16
2.3.3 Assign 'No Data' values 17
2.3.4 Confirm the settings to create the project 17
Lesson 3 More information out of DSM data: image filtering 18
3.1 Strategy to get more information out of DSM data 18
3.2 Translating the strategy – use 'image layer operation' algorithms 21
3.2.1 Introduction to the algorithm 'convolution filter' 21
3.2.2 Create a smoothed DSM layer using the 'convolution filter' algorithm 23
3.2.3 Create a slope layer using the 'surface calculation' algorithm 25
3.3 Review results– two new image layers are created 27
Lesson 4 Creating image objects 28
4.1 Strategy to create image objects – use slope layer and 'contrast split segmentation' 28
4.2 Translate the strategy – use algorithm 'contrast split segmentation' 30
4.2.1 Introduction to the algorithm 'contrast split segmentation' 30
4.2.2 Create initial objects using 'contrast split segmentation' 32
4.2.3 Review intermediate result 34
4.2.4 Refine with a second 'contrast split segmentation' step 35
4.3 Review result – medium and steep objects are separated 37
Lesson 5 Classifying steep areas 38
5.1 Strategy to classify objects with steep slope 38
5.2 Translating the strategy – use 'Mean of dsm_slope' 39
5.2.1 Find a threshold to classify steep areas 39
5.2.2 Insert process to classify steep objects 39
5.3 Review results – steep areas are classified 41
Lesson 6 Classifying the ground 42
6.1 Strategy to classify the ground – measure the lowest elevation 42
6.2 Translating the first step of strategy – compute a variable 44
6.2.1 Statistical operation type 'quantile' 44
6.2.2 Compute the 10th quantile of 'Mean DSM' 44
6.2.3 Execute the process and check the measured value 46
6.3 Translating the second step of strategy – use 'Min. pixel value' 46
6.3.1 Check the automatically calculated threshold for feature 'Min. pixel value' 46

6.3.2 Classify using the variable 47
6.4 Review result – ground is classified 49
Lesson 7 Classifying buildings 50
7.1 Strategy to classify buildings - the difference in elevation 50
7.2 Translating the strategy – use difference in elevation 51
7.2.1 Create 'Customized Relational Feature' 51
7.2.2 Find threshold for 'Building' classification 53
7.2.3 Insert process using the created feature and the algorithm 'assign class' 53
7.3 Review result – basic building shapes are classified 55
Lesson 8 Refining the 'Building' classification 56
8.1 Strategy to refine 'Building' classification - 'Area' feature 56
8.2 Translating strategy – use 'Area' feature for refinement 57
8.2.1 Find a threshold for de-classifying too small 'Building' objects 57
8.2.2 Insert process to refine classification by using the 'Area' feature 57
8.3 Review result – too small 'Building' objects are declassified 59
Lesson 9 Cleaning up 'Steep' objects 60
9.1 Strategy to reshape objects 61
9.2 Translating strategy - reshape using the spectral layers 62
9.2.1 Prepare process structure 62
9.2.2 Insert the 'multiresolution segmentation' process as a child process 62
9.3 Review result – objects are reshaped 64
9.4 Strategy to grow into spectrally similar objects 65
9.4.1 Concept of 'temporary classes' 65
9.4.2 Spectral similarity features 66
9.5 Translating strategy – classify adjacent, spectrally similar objects 67
9.5.1 Classify 'Steep' objects directly adjacent to 'Building' objects' as 'Building Candidates' 67
9.5.2 Create the feature 'Customized Brightness' 68
9.5.3 Excursus: Distance calculation in eCognition 70
9.5.4 Set the distance calculation mode in the 'set rule set options' algorithm 72
9.5.5 Create the customized relational feature expressing spectral differences/similarities 73
9.5.6 Find a threshold to classify spectrally similar 'Building Candidates' to the class 'Buildings' 74
9.5.7 Classify using the Customized Class-Related feature 75
9.5.8 Classify the temporary class 'Building Candidates' back to the class 'Steep' 77
9.6 Review the result – spectrally similar objects are classified 79
9.7 Strategy to fill holes in 'Building' classification 80
9.7.1 'Looping' a process – automatically repeat the procedure 80
9.7.2 Find totally enclosed objects 81
9.8 Translating strategy – find surrounded and enclosed 82
9.8.1 Classify those with a high relative border to 'Building' objects and loop the process 82
9.8.2 Review intermediate result 84
9.8.3 Classify those totally surrounded using algorithm 'find enclosed by class' 85
9.9 Review result – enclosed objects are classified 88
Lesson 10 Cleaning up 'Building' objects 90
10.1 Strategy to clean up vegetation misclassified as 'Building' 90
10.2 Translating 1st step: use spectral feature to declassify vegetated areas 92
10.2.1 Load the customized feature 'CustRatioGreen' 92

10.2.2 Create a process to declassify vegetation misclassified as 'Building' objects 92
10.3 Review result of first step – the big vegetation objects are gone 95
10.4 Strategy to get rid of again enclosed 95
10.5 Translating 2nd step – copy and paste existing 'find enclosed by class' process 96
10.6 Review result 96
10.7 Strategy to declassify too small objects 97
10.8 Translating 3rd step: declassify the too small 98
10.8.1 Create process to merge all 'Building' objects 98
10.8.2 Use the 'Image Object Table' tool to review the smallest objects 99
10.8.3 Sort the values and browse to the individual objects 100
10.8.4 Insert process to refine classification by using the 'Area' feature 101
10.9 Review result 104
Lesson 11 Preparing the export 106
11.1 Strategy to export the results – declassify those classes not of interest, merge objects 106
11.1.1 Clean up classification 106
11.1.2 Get rid of small objects 107
11.2 Translating strategy – declassify 'Steep' and 'Ground' 108
11.2.1 Prepare the Rule Set structure 108
11.2.2 Insert process to declassify 108
11.2.3 Create process to merge all 'unclassified' objects 108
11.3 Review result 110
11.4 Translating strategy – remove small objects 111
11.4.1 Use the 'Image Object Table' tool to review the smallest objects 111
11.4.2 Use 'remove objects' algorithm to "swallow" small objects 112
11.5 Review result 113
Lesson 12 Exporting the vector file 115
12.1 Insert process to export shape file with attributes 115
12.1.1 Create the feature 'Class Name' 115
12.1.2 Insert process to export vector file 116
12.2 Review the exported result 118
Where to get additional help and information? 120
eCognition Community and Rule Set Exchange platform 120
User Guide and Reference Book 120
Additional Guided Tours and Tutorials 121
eCognition Training 121
Consulting 121
Buy Software and Services? 121

Introduction

About this Guided Tour

Welcome to the Guided Tour **Advanced example of building extraction using LiDAR.**

This tour is written for **intermediate users** of the eCognition software. It is expected that the user has basic knowledge about the various eCognition products. It is recommended to work through the Guided Tour **Getting started – Example: Simple building extraction using LiDAR** before.

This Guided Tour will focus on the **more advanced steps** and strategies in developing a Rule Set using **eCognition**.

Further information about eCognition products is available on the website (http://ecognition.com).

Requirements

To perform this tutorial you will need

- **eCognition Developer** installed on a computer.

All steps of this Guided Tour, can be done using **eCognition Developer or its trial version** (http://ecognition.com/products/trial-software).

This edition is designed for self-study.

Data Included with the Guided Tour

Image Data

Image data can be found in the folder **…/Data**.

There are two **.img** files available for this Guided Tour, 'RGB_Level3_Advanced_Example.img' representing the aerial RGB data and 'DSM_Level3_Advanced_Example.img' representing the LiDAR Data.

Please note that the geoinformation of the data has been deleted.

Data courtesy of Woolpert, a premier supplier of ADS40 imagery and LiDAR data

Rule Sets

Rule sets are available in the **Rule Sets** folder. Whenever the Guided Tour refers to a Rule Set it is to be found in this folder.

Project

The ready set up project can be found in the folder **project**.

Import Template

Import Templates can be found in the folder '**Data**'. Whenever the Guided Tour refers to an import template it is to be found in this folder.

How to use this Guided Tour

Symbols at the side of the document

If the side is hachured and 'Introduction' is added, this indicates that the text is giving a general introduction or methodology about the following Lesson, method or exercise.

Introduction

If the side is hachured and 'Information' is added, this indicates that the text is giving information about the following exercise.

Information

If this symbol is shown, you have to follow the numbered items in the text. If you just want to work through the exercises without reading the theory part, follow only this sign.

Action!

If this symbol is shown, compare the settings shown in the screenshot with the settings in the according dialog box in Developer.

Settings Check

If this symbol is shown check the screenshot of the Process Tree with the content of the Process Tree in Developer.

Rule Set Check

If this symbol is shown check the screenshot aside with the result Developer. It should look similar.

Result Check

Symbols for next step in Rule Set development

The following symbols show you always the next step to do. They will be prompted in the text.

Get the big picture

This diagram expresses that the next step to do is to develop the core strategy, containing the choice of the data to use. Get the big picture:

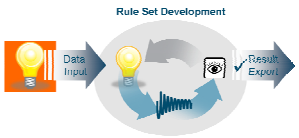


Figure 1: Get the big picture

Import data

This diagram expresses that the next step to do is to import your data:

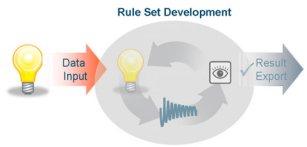


Figure 2: Import data

Develop strategy

This diagram expresses that the next step to do is to develop your next Rule Set Development strategy step:

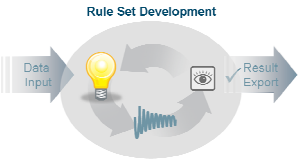


Figure 3: Develop Strategy

Translate strategy into Rule Set

This diagram expresses that the next step to do is to translate your strategy into a Rule Set:

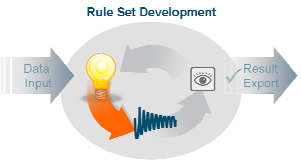


Figure 4: Translate into Rule Set

Review result

This diagram expresses that the next step to do is to review your results:

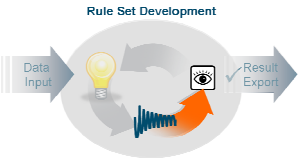


Figure 5: Review Result

Refine/expand strategy

This diagram expresses that the next step to do is to go back and develop your next strategy step, because the result did not meet your requirements:

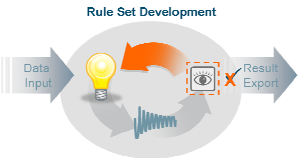


Figure 6: Refine/Expand Strategy

Ready for export

This diagram stands for the stage where you are satisfied with the result and you can export:

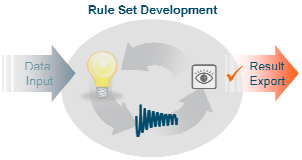


Figure 7: Result is OK, ready for export.

Symbols for stage in Rule Set development

There are also diagrams in the header of the document showing the current stage you are:

Get the big picture

Import the data

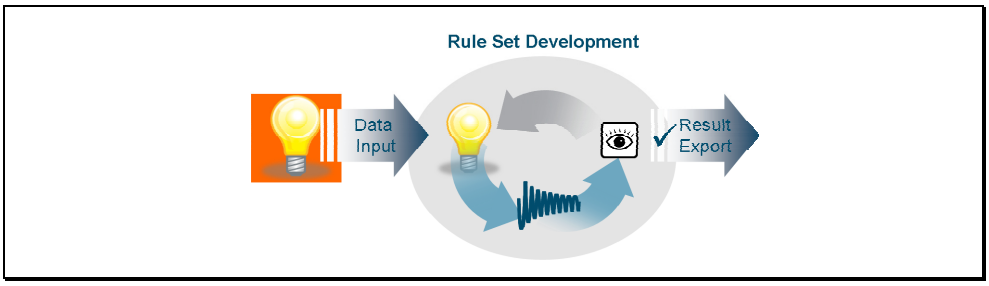
Develop strategy step

Write Rule Set

Review intermediate result

Export result

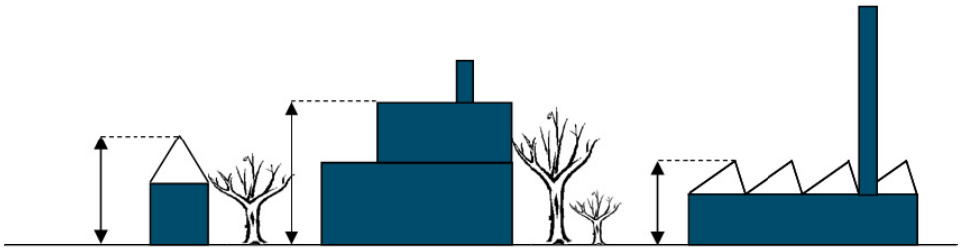
Lesson 1 Get the big picture



Classifying buildings using fixed heights, like in the Guided Tour 1 is not very transferable. The height can differ from scene to scene.

In this advanced Guided Tour you will learn techniques to classify buildings in a much more transferable way. Therefore first robust and significant features of buildings have to be identified.

One very robust characteristic of buildings is the **sudden change in elevation** at its borders and that buildings have a very **different elevation** than their surrounding.



So on the one hand the high slope surrounding a building and on the other their difference in height are used to classify buildings.

For this module a larger subset with more complex buildings is used.

Creating image objects

- A new image layers will be created, using smoothing and surface filters, additionally a buffer at scene border is created.

Figure 8: New created image layer, representing information about slope.

- The created slope layer has high contrasts. The steep areas and flat areas are analyzed. Therefore a segmentation on basis of the slope layer using the 'Contrast Split Segmentation' is applied. This segmentation gives back the best result in one step.

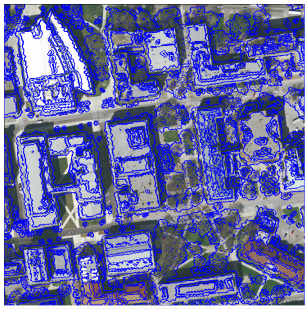


Figure 9: Image objects are created on basis of the new created layer using the auto adaptive algorithm 'contrast split segmentation'.

Classification

Identify those objects, which are obviously not buildings

The overall classification approach to get initial building classification is a so called 'masking technique'. This means that objects are excluded which obviously do not belong to the target class 'Building'.

- Classify steep areas

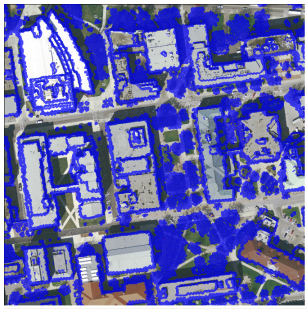


Figure 10: Steep areas are classified (blue).

- Classify ground

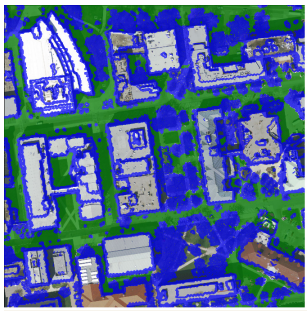


Figure 11: Ground is classified (green).

Classify remaining unclassified as buildings

- Classify buildings based on their difference in elevation.

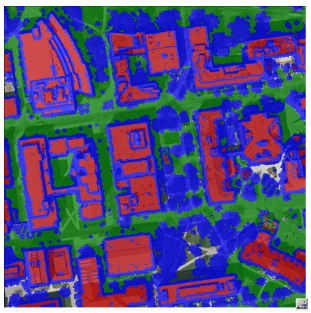


Figure 12: Buildings are classified (red).

Refine the buildings by growing them into steep areas

- Reshape objects based on spectral layers and refine buildings based on context, shape and spectral features.

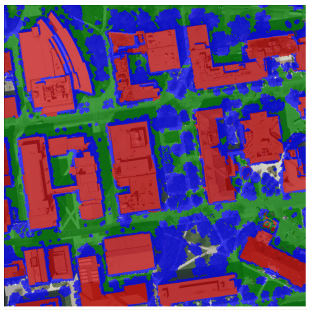
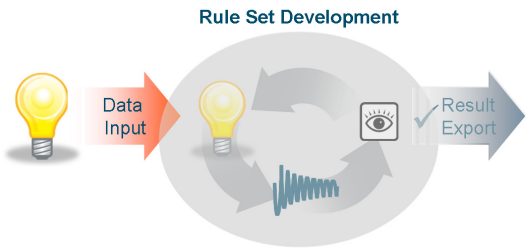


Figure 13: Buildings (red) are refined.

Export the result

- Export a shape file.

Next step: Import the data!



Lesson 2 Loading the data

This lesson covers the following content

- *Open eCognition Developer Rule Set Mode*
- *Option 1: Using workspace and import template*
- *Option 2 (Trial version): Create an individual Project*

Information

For this module a larger subset with more complex buildings is used.

The scene to be loaded has an extend of 2005 to 2005 pixel, this means **405002**5 pixels. The resolution of the RGB data is **0,5 feet**, respectively 0.1524 meter. This means an area of 1012506 square inches, respectively almost one square kilometer is analyzed.

NOTE:

If you are working with the Trail Version, you have to jump to chapter 2.3 to **create** and **save** projects individually. The workspace functionality is not available.

2.1 Open eCognition Developer Rule Set Mode

Introduction

When starting eCognition Developer you can optionally start it in either:

- Quick Map Mode

or

- Rule Set Mode

The Quick Map Mode is designed to allow a user **solving simple analysis tasks** without having to get involved with Rule Set development. The main steps in analyzing an image are creating objects, classifying objects and exporting results. For each of these steps, a **small assortment of predefined actions** is available.

The **Rule Set Mode** provides all necessary tools to **develop own Rule Sets**. In all the Guided Tours, the Rule Set Mode is necessary to accomplish the exercises.

Action!

1. Start eCognition Developer.
2. Select 'Rule Set Mode'.

Settings Check

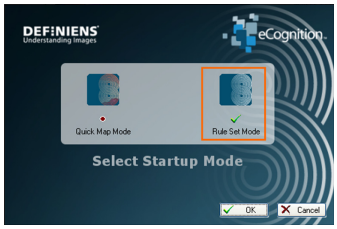


Figure 14: Select 'Rule Set Mode' to have all tools available.

2.2 Option 1: Using workspace and import template

2.2.1 Create a new workspace

Information

In the 'View Settings' toolbar there are **4 predefined View Settings** available. Each view setting is specific to the different phases of a Rule Set development workflow. To create, open or modify a workspace, make sure that you are in the 'Load and Manage Data' view.

Action!

1. Select the predefined view setting number 1 '**Load and Manage Data'** from the 'View Settings' toolbar.
2. To **create a new workspace**, do one of the following:
   - Click the '**Create New Workspace button**' on the toolbar.
   - Choose '**File > New Workspace**' from the main menu bar.

The 'Create New Workspace' dialog box opens.

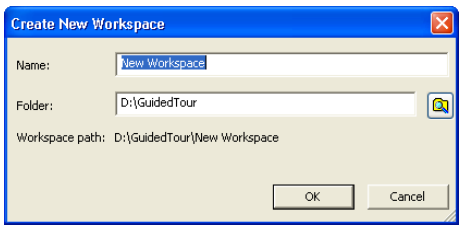


Figure 15: 'Create New Workspace' dialog box.

3. **Enter a Name** for the new workspace. The default name is 'New Workspace'.
4. Browse for a **location to save** the new workspace.
5. Confirm with '**OK**' to create the workspace. It is displayed as the root folder in the left pane (the tree view) of the workspace window.

2.2.2 Import data in the workspace with an import template

Information

In this Guided Tour the project used contains an RGB image file plus the LiDAR DSM.

Every channel, **red, green, blue** and the **DSM** shall have an **alias** to make further processing more transferable. This means the loaded data will not appear with its default naming Layer1, Layer2 etc. but have meaningful names like red, green, blue and DSM.

Besides that areas containing **no data values** have to be defined.

All these settings are already defined in the customized import template file (in **xml format**) we want to use for import.

Action!

1. Select the predefined view setting number 1 '**Load and Manage Data'** from the 'View Settings' toolbar.
2. Right-click in the left workspace window and choose 'Customized Import' from the menu.

The 'Customized Import' dialog box opens.

3. Make sure that the 'Workspace' tab is active.
4. Click on the 'Load' button and browse to the '…\GuidedTour3_eCognition8_AdvancedRuleSetWriting_ExampleBuildingExtraction**Data** folder. Select the '**CustoImport_GuidedTour_Level3.xml**'.
5. Define the 'Root Folder' by browsing to the folder the data is stored.
6. Define '**RGB_Level3_Advanced_Example.img**' as Masterfile.
7. Confirm with 'OK'.

Settings Check

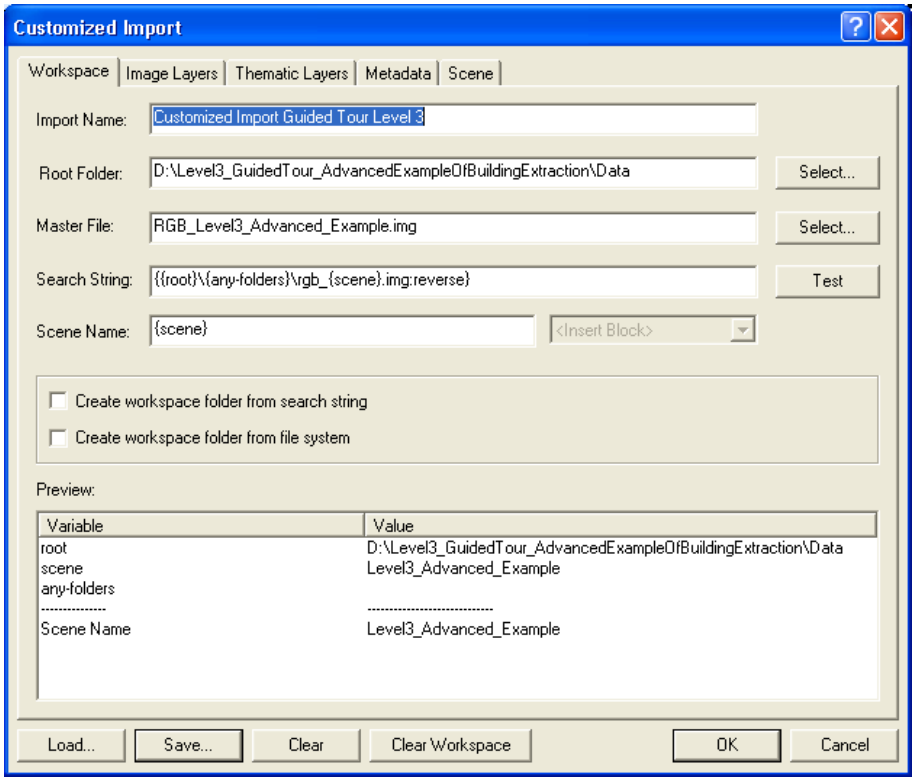


Figure 16: The settings defined in the .xml are loaded

Check Workspace

Figure 17: The project 'Level3_Advanced_Example' is created.

2.3 Option 2 (Trial version): Create an individual Project

2.3.1 Define the data to be loaded

Action!

1. To create an individual project, click on the '**Create New Project**' button or go to main menu 'File>New Project'

The 'Create Project' dialog box together with the '**Import Image Layer**' dialog box opens.

2. Navigate to the folder '…\GuidedTour3_eCognition8_AdvancedRuleSetWriting_ExampleBuildingExtraction**Data**'.

3. Mark the following image files and click '**Open**'.

   - **RGB_Level1_Simple_Example.img04mar_multi.img**
   - **DSM_Level1_Simple_Example.img**

The '**Create Project**' dialog box opens.

4. In the 'Create Project' dialog box in the field 'Name' either enter a meaningful name for the project, e.g. '**Building extraction**'. Or keep the default naming according to the first loaded image file 'RGB_Level1_Simple_Example'.

2.3.1 Define layer alias

Information

In order to generate Rule Sets that are transferable between different datasets, the loaded channels have to have aliases assigned to them.

Action!

1. To assign a layer alias, select the **layer** in the 'Create Project' dialog box and **double-click** it.

2. Assign for layer 1 'blue'.

3. Confirm the alias with 'OK'

Assign the following **aliases** to the other layers,:

- **Layer1 →red**
- **Layer2 →green**
- **Layer3 →blue**
- **Layer4 →DSM**

2.3.2 Assign 'No Data' values

If there are areas with 0 or other artificial values not containing any relevant information, they can be set as 'No Data' areas. These areas will not be processed in any way, no objects will be created for these areas. In the current example, there are areas, where the RGB contains no information but the DSM, these areas are set to be 'No Data'.

Information

4. Click on the '**No Data**' button.
5. In the field '**Global No Data Value**' switch on the check-box and enter the value 0.

Action!

This indicates that an area is defined as 'No Data' if any of the layers contains a 0 value pixel.

6. Confirm the settings with 'OK'.

2.3.3 Confirm the settings to create the project

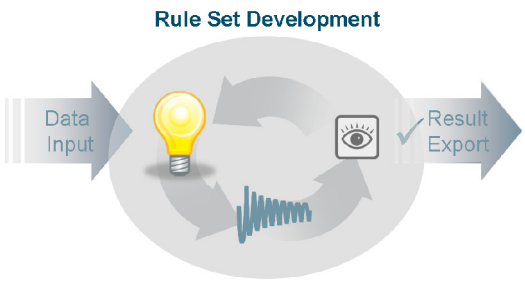
7. After all alias are defined, click '**OK**' at the bottom of the 'Create Project' dialog box.

The new project is created.

Lesson 2 covered the following content

- → *Open eCognition Developer Rule Set Mode*
- → *Option 1: Using workspace and import template*
- → *Option 2 (Trial version): Create an individual Project*

Next step: Develop Strategy for fist analysis step!



Lesson 3 More information out of DSM data: image filtering

This lesson covers the following content

- *Strategy to get more information out of DSM data*
- *Translating the strategy – use 'image layer operation' algorithms*
- *Review results– two new image layers are created*

3.1 Strategy to get more information out of DSM data

Around a building the slope is obviously very **steep**. Leaf-off trees, which are elevated objects as well, are totally covered by steep slopes, due to the fact that the lasers one time hits a branch, the other time goes through to the ground.

This information is quite unique and can be used as characteristic feature for classification. But this slope information is not directly contained in the data loaded.

eCognition Developer provides the possibility get more out of your image layers by running **filter algorithms** to creating new layers. This can be **smoothing** filters to reduce noise/detail or **edge filters**.

A very interesting filter for building extraction is the '**Surface calculation**' filter, which calculates slopes within the data.

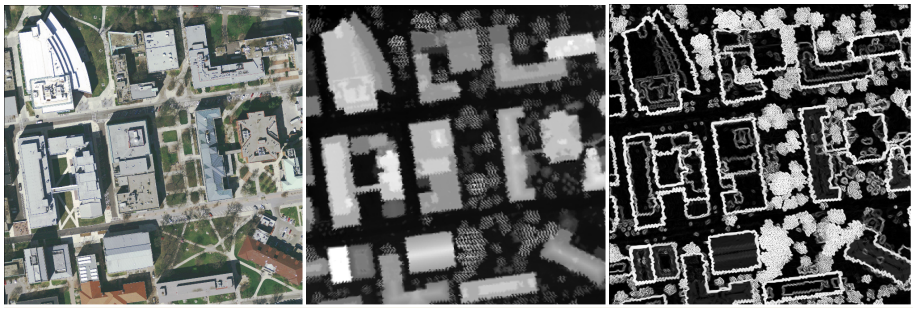


Figure 18: Left: RGB data; Middle: DSM; Right: new created slope layer

Differences in resolution between DSM and RGB lead to saw-tooth distortion

The different resolutions of DSM and RGB data result in artificial, wrong slope calculation, when creating applying a slope filter on the raw, DSM layer which has a lower resolution then the loaded RGB data.

The effect from differing resolutions can be avoided by:

- First creating a **smoothed** DSM layer
- Second create on basis of the smoothed DSM the **slope** layer.

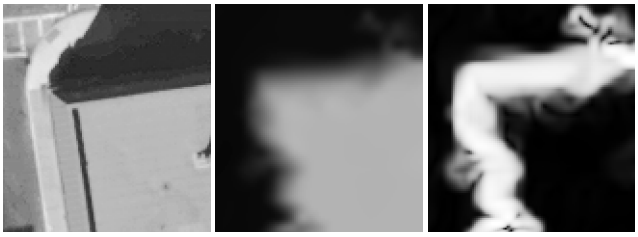
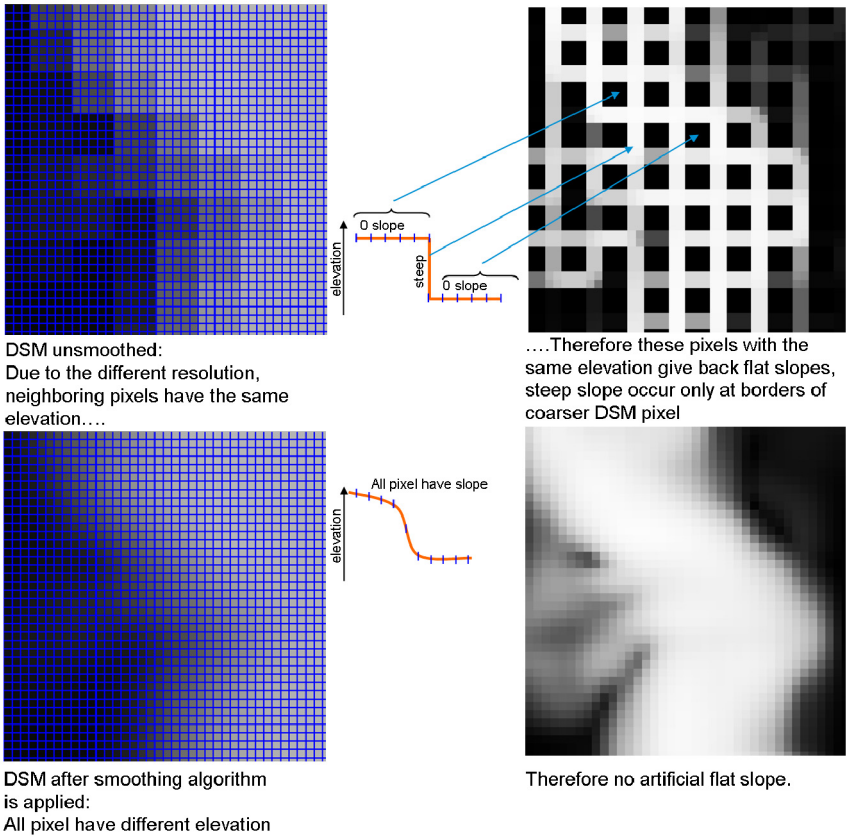


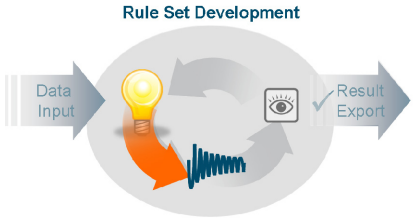
Figure 19: Left: Red image layer, Middle: DSM smoothed; Right: Slope filter applied on smoothed DSM.

The graphic below shows the difference in slope calculation using the unsmoothed or the smoothed DSM:



Consequently two different algorithms have to be used. One creating a smoothed DSM layer and one creating a slope layer based on the smoothed DSM layer.

Next step: Translate Strategy into Rule Set!



3.2 Translating the strategy – use 'image layer operation' algorithms

This chapter covers the following content

- ➔ *Introduction to the algorithm 'convolution filter'*
- ➔ *Create a smoothed DSM layer using the 'convolution filter' algorithm*
- ➔ *Create a slope layer using the 'surface calculation' algorithm*

3.2.1 Introduction to the algorithm 'convolution filter'

Introduction

The 'convolution filter' algorithm uses the **Gauss Blur** filter for smoothing. Important setting is the **Kernel size**. The Kernel size influences the degree of smoothing, as it defines the box with which the individual pixel is compared. The higher the Kernel size, the more the data is smoothed.

In our case, the resolution of the RGB data is 5 times lower than the resolution of the DSM, this means the Kernel must be about 10. The value 9 already gives back the desired result.

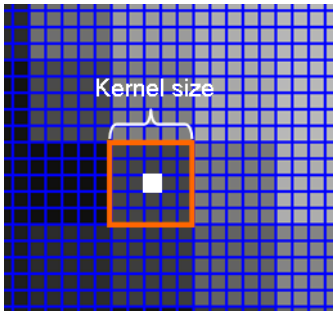


Figure 20: Diagram of Kernel size.

Use the **surface calculation** algorithm to derive the slope for each pixel of a digital elevation model (DEM). This can be used to determine whether an area within a landscape is flat or steep and is independent from the absolute height values.

Two algorithm types are available:

- Slope Zevenbergen , Thorne (ERDAS):

Uses the Zevenbergen Thorne method to calculate slope. See: Quantitative analysis of land surface topography. Zevenbergen, L W; Thorne, C R Earth Surface Processes and Landforms [EARTH SURF. PROCESS. LANDFORMS.], vol. 12, no. 1, pp. 47-56, 1987 #

- Aspect (Horn's Method)

Uses Horn's Method to calculate aspect. See: Horn, B. K. P. (1981). Hill Shading and the Reflectance Map, Proceedings of the IEEE, 69(1):14-47.

In this Guided Tour the '**Slope Zevenbergen, Thorne (ERDAS)**' algorithm is used. With this algorithm you can define:

Introduction

- Which '**Gradient unit**' to use, percent or degree. In this Guided Tour the 'Percent' is used.
- The '**Unit of Pixel Values**' Here the unit of the pixels compared to the unit of the elevation is set. In our case the value 1 is kept. This will give back a wider range of percentages. Respectively the slope is strengthened and more contrasted. The values will range between 0 and 192 percent not from 0 to 128 percent.
- As '**Input layer**' the smoothed DSM layer is used.

3.2.2 Create a smoothed DSM layer using the 'convolution filter' algorithm

Prepare the process structure

1. Select the predefined view setting number 4 '**Develop Rulesets**' from the 'View Settings' toolbar.

2. **Right-click** in the 'Process Tree' window and select '**Append New**' from the context menu. In the 'Name' field enter the name '**Building Extraction**' and confirm with '**OK**'.

3. **Right-click** on the process 'Building Extraction' and select '**Insert Child**' from the context menu. Insert '**Apply Image Filters**' in the 'Name' field and confirm with '**OK**'.

Action!

Insert process using algorithm 'convolution filter'

Information

The 'convolution filter' uses the Gauss Blur filter for smoothing. Important is to set the Kernel size, as discussed in 3.2.1 Introduction to the algorithm 'convolution filter'.

1. **Right-click** on the parent process '**Apply Image Filters**' and select '**Insert Child**' from the context menu.

Choose the algorithm

The 'Image layer operation' algorithms are not part of the default algorithm list. You have to add it to the drop down list first.

2. Drop-down the algorithm list.

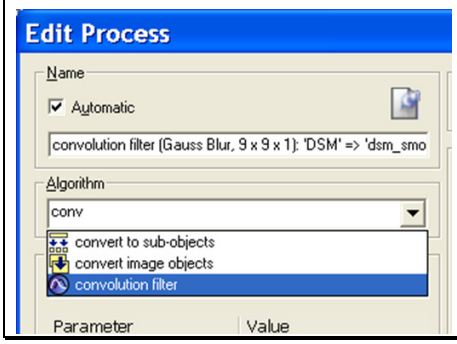
3. Scroll to the end of the list and click on '**more…**'.

4. In the 'Available algorithms' window, go to the category '**Image layer operation**' and double-click on '**convolution filter**' and '**surface calculation**' too.

Action!

NOTE:

Instead of picking algorithms from the drop-down list, you can simply type in the first two or three letters of the algorithm and you get a selected list.



Both algorithms are now listed in the 'Selected algorithms' window.

5. Confirm with '**OK**'.
6. Select the algorithm '**convolution filter**' in the algorithm drop-down list.

Settings Check

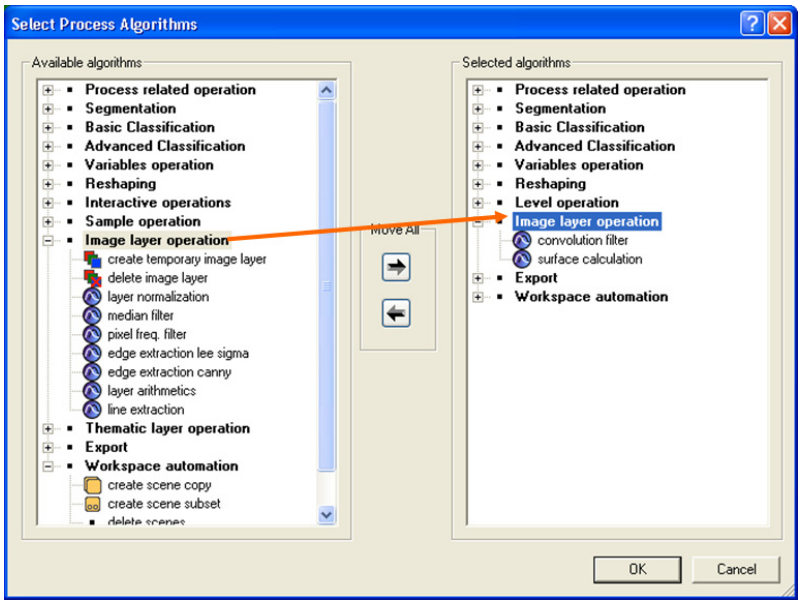


Figure 21: The algorithms convolution filter' and 'surface calculation' are moved to the 'Selected algorithms' list.

Set the algorithm parameter

Action!

1. In the field 'Type' keep '**Gauss Blur**', keep the 'Advanced parameter' 1.
2. In the field' **2D kernel size**' insert **9**.
3. In the field '**Input layer**' define the source layer for smoothing: '**DSM**'.
4. In the field '**Output layer**' type in '**dsm_smoothed**'. This will be the name of the smoothed layer to be created.
5. In the field '**Output layer type**' keep '**as input layer**'.
6. Confirm the settings with '**OK**'.
7. Right-click on the process and select '**Execute**' from the context menu.

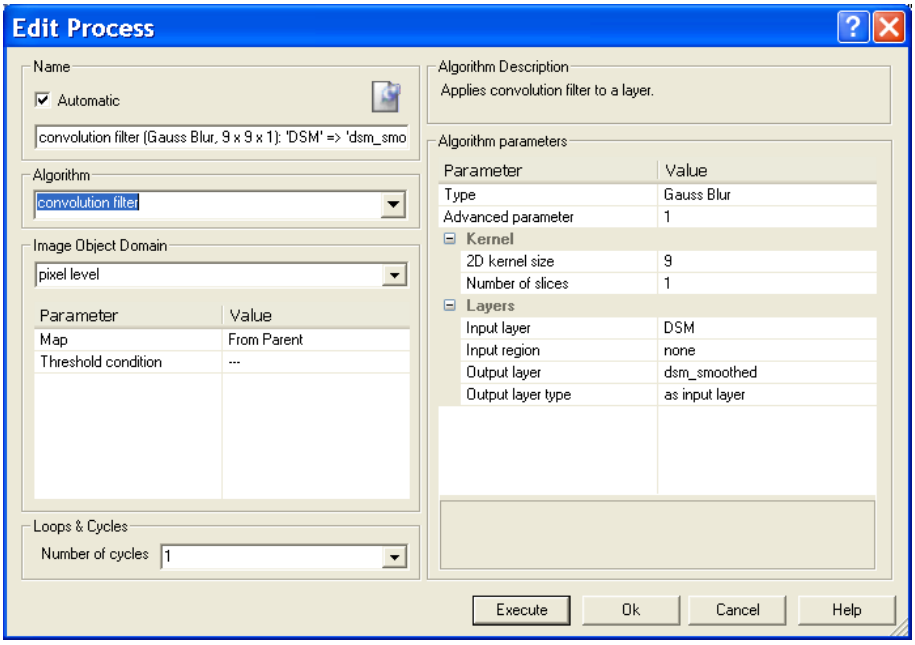


Figure 22: Process settings to create new layer 'dsm_smoothed'.

Settings Check

3.2.3 Create a slope layer using the 'surface calculation' algorithm

1. Right-click on the last process and select 'Append New' from the context menu.
2. Select the algorithm '**Surface calculation**' in the 'Algorithm' drop own list.

Action!

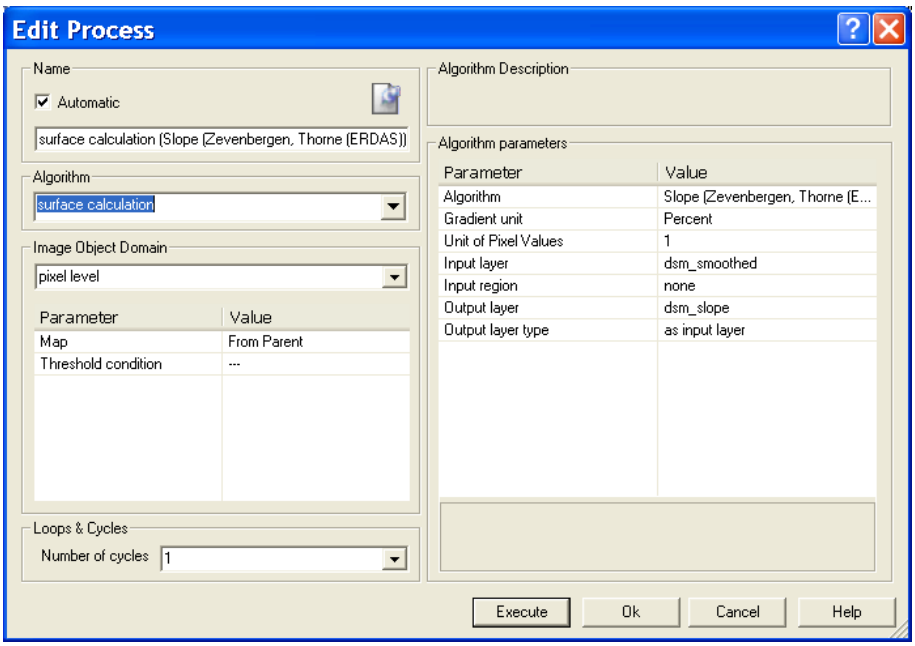


Figure 23: Process settings to create new layer 'dsm_smoothed'.

Settings Check

Set the algorithm parameter

Action!

1. In the field 'Algorithm' keep 'Slope Zevenbergen , Thorne (ERDAS)'
2. In the field '**Gradient unit'** keep '**Percent**'.
3. In the field' Unit of Pixel Values' keep 1.
4. In the field '**Input layer**' choose '**dsm_smoothed**' from the drop-down list.
5. In the field '**Output layer**' insert '**dsm_slope**'. This will be the name of the layer to be created.
6. Confirm the settings with 'OK'.
7. Right-click on the process and select '**Execute**' from the context menu.

Rule Set Check

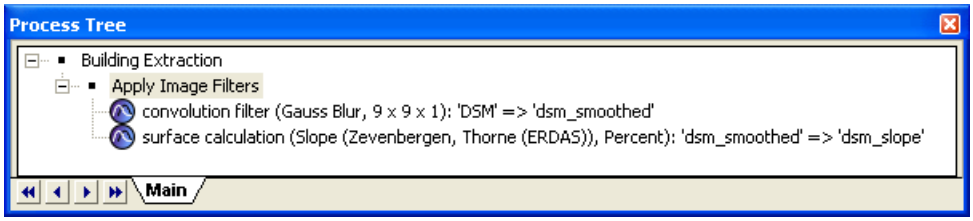
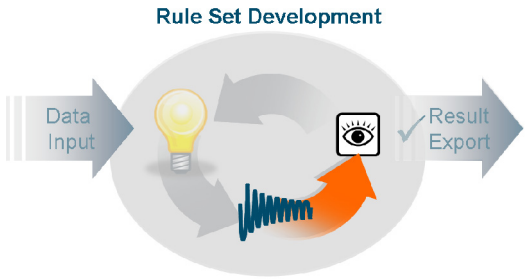


Figure 24: Process Tree with image layer operation processes added.

Next step: Review Result!



3.3 Review results– two new image layers are created

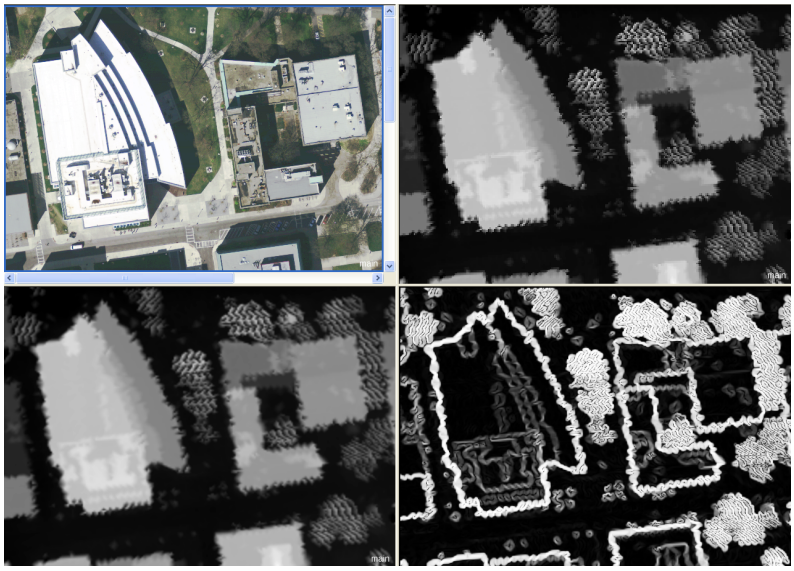


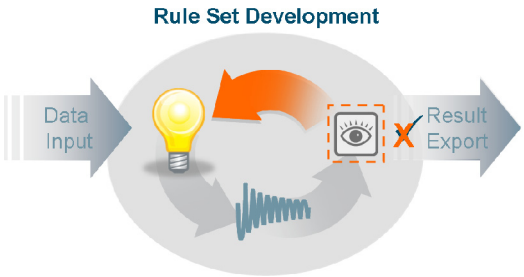
Figure 25: Top left: the optical RGB data; Top right: the original DSM; Lower left: the smoothed DSM; Lower right: the slope layer.

Result Check

Two additional image layers have been created, the smoothed DSM and the slope layer, which is based on the smoothed DSM.

The '**smoothed_DSM**' layer is basis for later segmentation and classification.

Next step: Develop next analysis step!



Lesson 3 covered the following content

- *Strategy to get more information out of DSM data*
- *Translating the strategy – use 'image layer operation' algorithms*
- *Review results– two new image layers are created*

Lesson 4 Creating image objects

This lesson covers the following content

- *Strategy to create image objects – use slope layer and 'contrast split segmentation'*
- *Translate the strategy – use algorithm 'contrast split segmentation'*
- *Review result – medium and steep objects are separated*

4.1 Strategy to create image objects – use slope layer and 'contrast split segmentation'

Introduction

For further classification objects are needed to represent the steep areas of the slope layer. The high slopes are very **high contrasted** in the slope layer.

In opposite to the other segmentation algorithms, the 'contrast split segmentation' is predestinated to segment images with significant contrasts, like it is the case for the slope layer, at once and auto adaptive. For example with the multiresolution segmentation an additional classification and merge steps would be necessary to get a similar result and it would not be auto adaptive.

Simply said, the '**contrast split segmentation**' splits up objects into **dark and bright**. It calculates the **best threshold** for splitting automatically. The 'contrast split segmentation' is fast and creates very suitable image objects representing in our case the slope information.

Introduction

As the contrast split segmentation works on existing objects only, within the algorithm first an object level is created using a very large tile size. After this the actual contrast split segmentation is then applied.

Two segmentation steps will be necessary, one to get the general steep initial slope objects, a second to separate those with medium slope from the very steep.

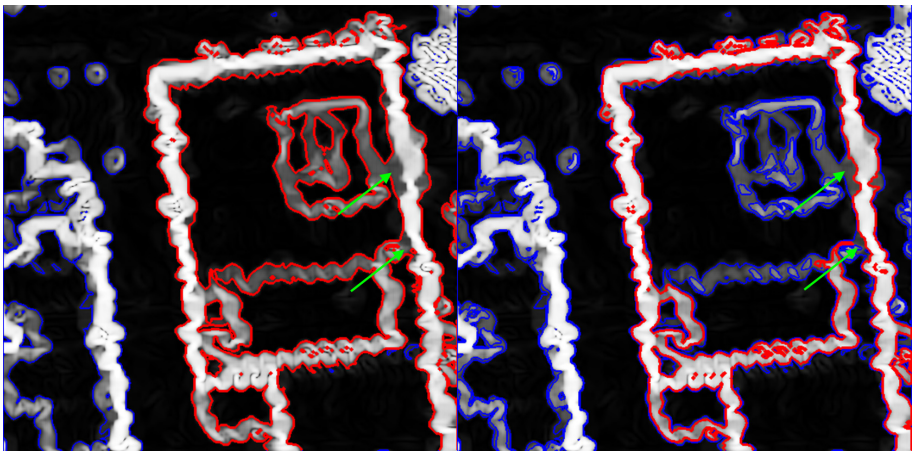
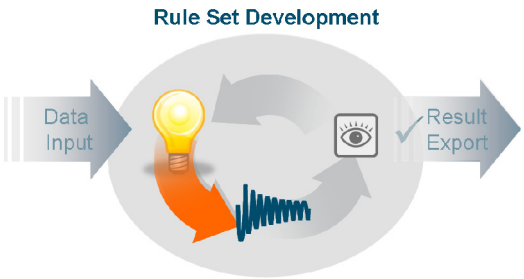


Figure 26: Left: Objects after first segmentation step; Right refined after second segmentation step.

Strategy summary:

- Create initial objects using 'contrast split segmentation
- Refine the object segmentation with a second 'contrast split segmentation' step

Next step: Translate Strategy into Rule Set!



4.2 Translate the strategy – use algorithm 'contrast split segmentation'

This chapter covers the following content

- ➔ *Introduction to the algorithm 'contrast split segmentation'*
- ➔ *Create initial objects using 'contrast split segmentation'*
- ➔ *Review intermediate result*
- ➔ *Refine with a second 'contrast split segmentation' step*

4.2.1 Introduction to the algorithm 'contrast split segmentation'

Introduction

The following things you have to consider for the set up:

- The tile size for the initial objects
- The value range
- The step-size to proceed from the lowest value to the largest
- The contrast mode after which the contrast is calculated

The tile size for the initial, internal object creation

The tile size should be large enough to ensure that only one object per scene is created. Assuming that no tiles/subsets are processed larger than **10000x10000 pixels**, the tile size is set to this value.

If a lower value is chosen, there could appear artificial object borders.

Evaluate the slope layer to get minimum and maximum threshold

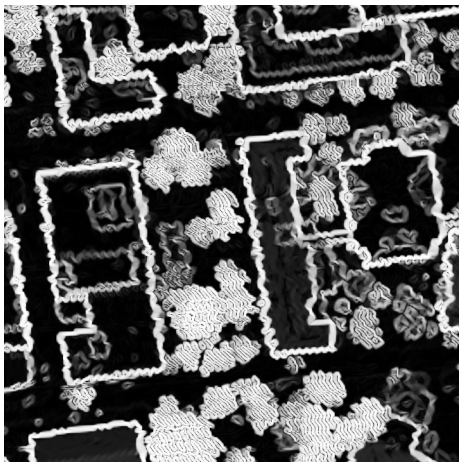


Figure 27: 'dsm_slope' layer.

1. Display only the slope layer. To do so use the 'Edit Layer Mixing' dialog box and switch off all other image layers.

Action!

Minimum and maximum value of the slope layer

2. Check the minimum and maximum values by creating the features '**Scene features>Scene related> Largest actual pixel value**' and '**>Smallest actual pixel value**' and '**Largest actual pixel value**'.

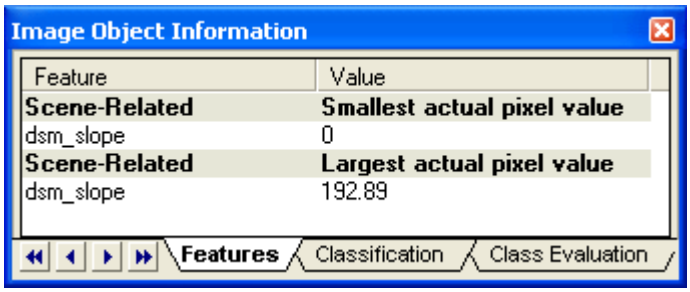


Figure 28: 'Image Object Information' window with smallest and largest actual pixel value displayed.

These values represent the steepest and the flattest slope, respectively the maximum and minimum value of the range to look into for contrast split segmentation.

Step size

Introduction

The step size defines by which value the threshold will increase from the **minimum threshold** to the **maximum threshold.** The value can either be added to the threshold or multiplied by the threshold, according to the selection in the '**Stepping type**' field.

The algorithm recalculates a new best threshold each time the threshold is changed by application of the values in the '**Step size**' and '**Stepping type**' fields, until the **maximum threshold** is reached.

Higher values entered for '**Step size**' will tend to execute more quickly; smaller values will tend to achieve a split with a larger contrast between bright and dark objects.

Conclusion:

In our example a **step size of 20** is suitable. A lower step size would increase processing time, but bring no better results, a coarse value would result into too generalized objects.

4.2.2 Create initial objects using 'contrast split segmentation'

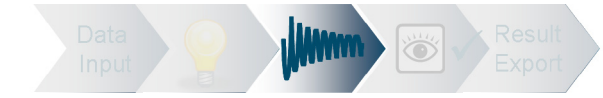
Action!

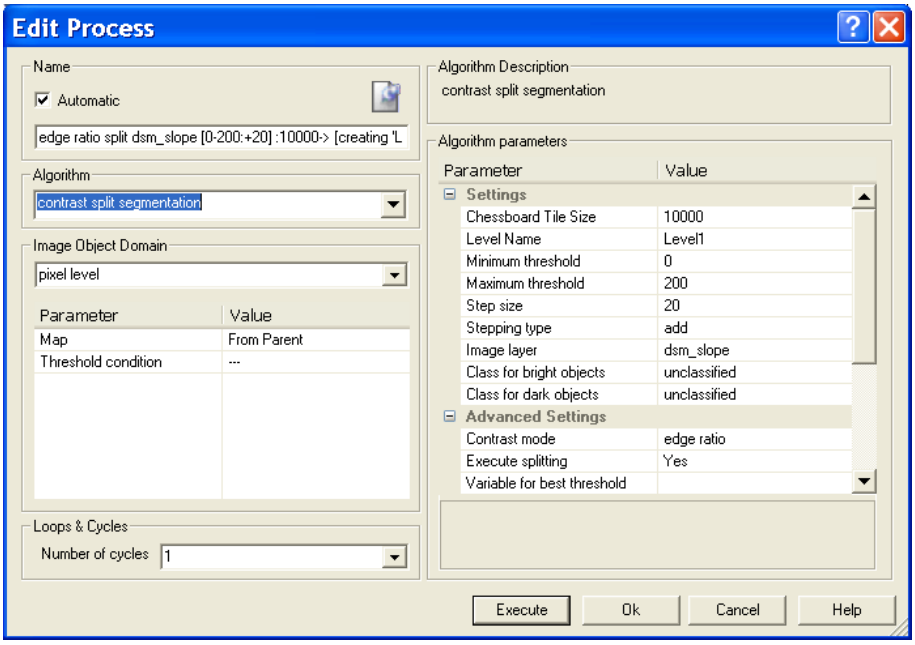
1. Collapse the parent process 'Apply Image Filters', right-click on it and select 'Append New'. Name the process '**Segmentation**'.
2. Right-click on the new parent process and select 'Insert Child' from the context menu.
3. Choose the algorithm '**contrast split segmentation**'.
4. In the 'Image Object Domain' keep from '**pixel level**'.

Define algorithm parameters

1. Set the '**Chessboard Tile Size**' to **10000**.
2. In the field '**Minimum threshold**' field insert 0.
3. In the field '**Maximum threshold**' field insert **200**.
4. In the field '**Step size' 20**.
5. Keep '**add**' in the field 'Stepping type'.
6. In the field 'Image Layer' choose '**dsm_slope**'. This specifies the image layer on which the segmentation will be based on.
7. Set '**unclassified'** in the fields '**Class for bright objects**' and '**Class for dark objects**'. These fields could be used if you immediately want to classify.
8. Keep the default settings of all other fields.
9. **Execute** the process.

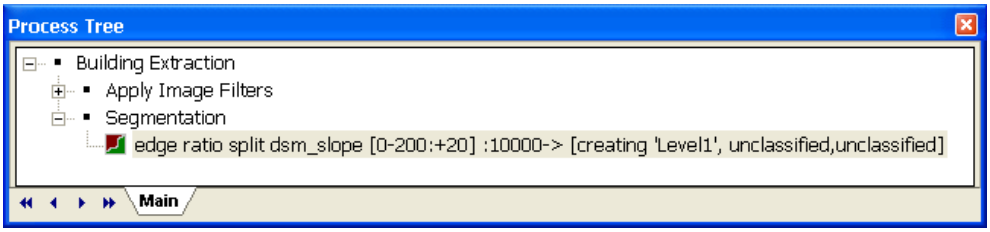
Level1 is created according to the settings in the process.





Settings Check

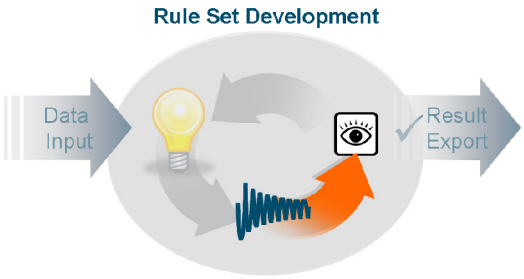
Figure 29: Process settings for creating image objects on basis of the 'dsm_slope' layer using the 'contrast split segmentation'.



Rule Set Check

Figure 30: Process Tree with first segmentation process added.

Next step: Review Result!



4.2.3 Review intermediate result

Result Check

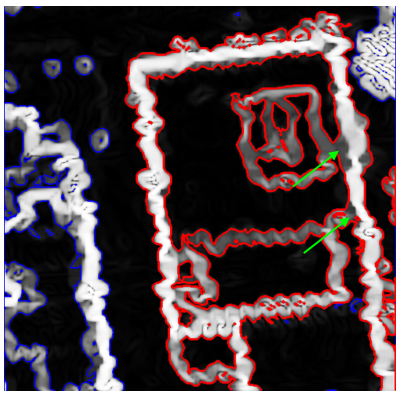
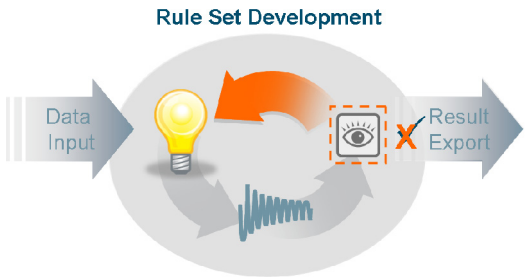


Figure 31: Objects after first segmentation step.

All contrasted objects have been segmented, but still medium and high slope are not separated yet.

Therefore **another segmentation step** must be added to separate those, again using the 'contrast split segmentation', but with **different settings**.

Next step: Develop strategy for next analysis step!



4.2.4 Refine with a second 'contrast split segmentation' step

Recap of Strategy:

Introduction

To **separate medium and very steep** areas, another segmentation step has to be added. Again the 'contrast split segmentation' is used, but with different settings.

Find new minimum value for contrast

Display the 'dsm_slope' and browse the cursor over areas where steep and very steep areas connect.

There medium steep areas have upper **values about 70**. From 70 onwards, the very steep areas begin.

Copy, paste and edit the 'contrast split segmentation'

Action!

1. Right-click on the 'contrast split segmentation' process and select 'Copy' from the context menu. 2. Right-click again and select 'Paste'.

An exact copy of the previous process has been added.

3. Double-click on the pasted process to open the 'Edit Process' dialog box.

Alternatively use Ctrl + C and Ctrl + V.

4. In the 'Image Object Domain' change from 'pixel level' to '**image object level**'.
5. Make sure that '**Level1**' is selected. , only objects already created in Level1 shall be reshaped.
6. In the field '**Minimum threshold'** change the value to **70**.
7. Confirm the settings with 'OK'.
8. **Execute** the process.

Settings Check

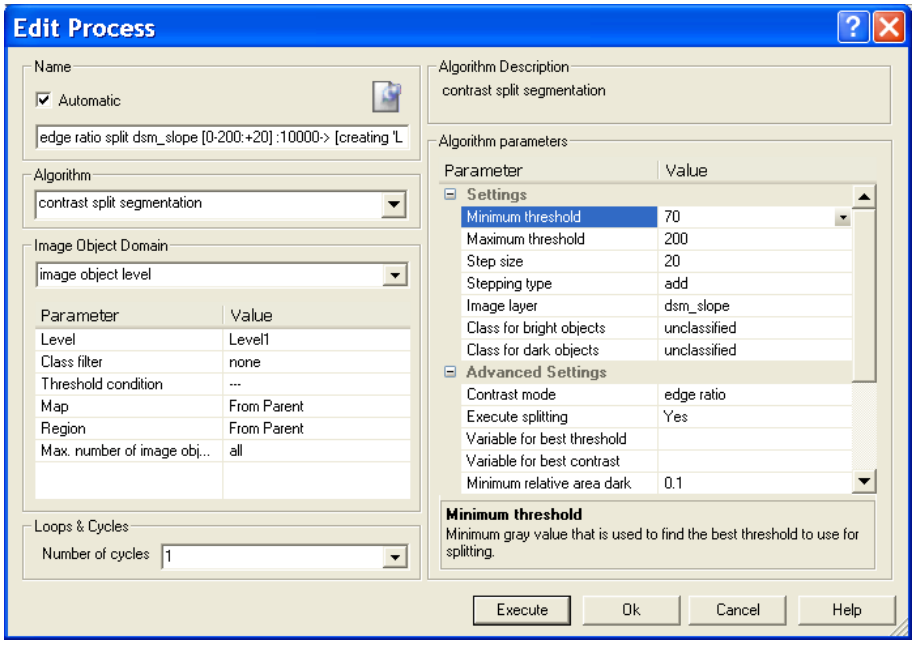


Figure 32: Process settings for refine objects using the 'contrast split segmentation'.

Rule Set Check

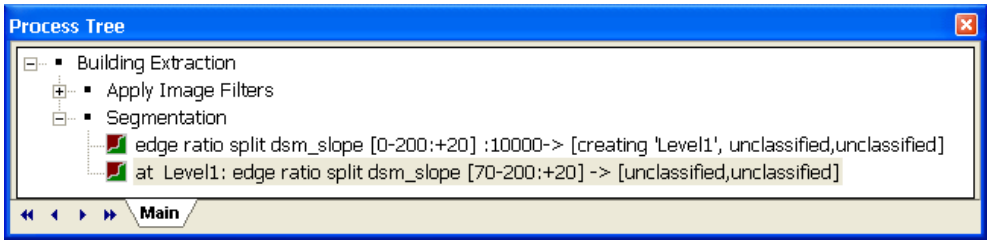
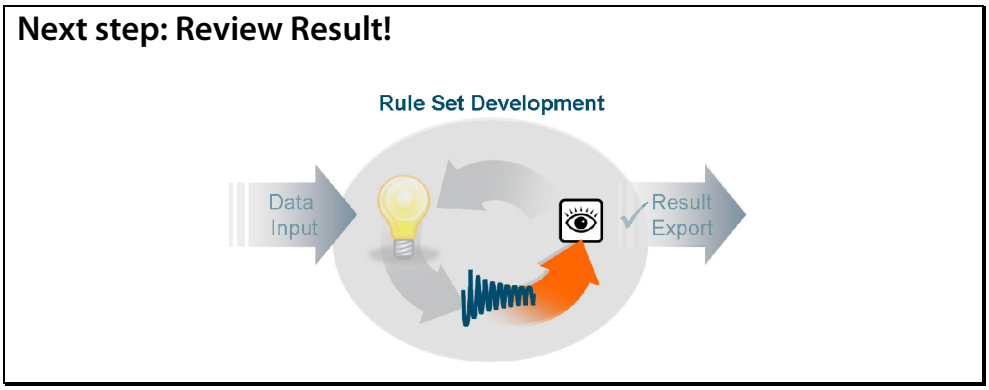


Figure 33: Process Tree with second segmentation process added.



4.3 Review result – medium and steep objects are separated

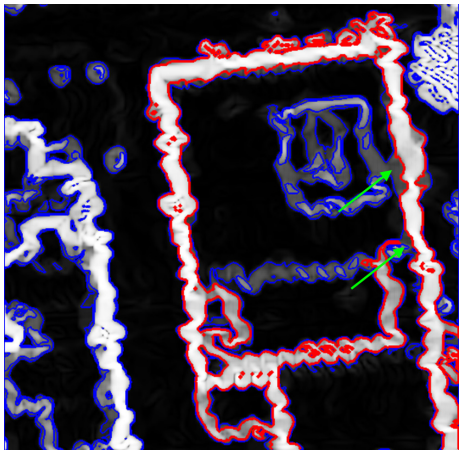


Figure 34: Objects after the second segmentation step.

Settings Check

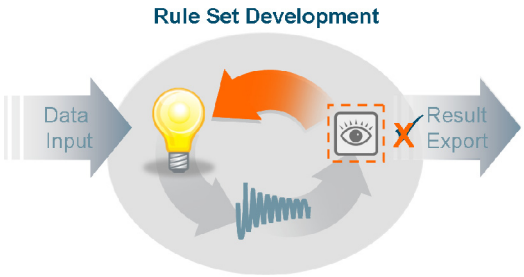
The objects have been split up into areas of very steep slope and areas with medium steep slope.

The objects are now ready for classification.

Lesson 4 covered the following content

- *Strategy to create image objects – use slope layer and 'contrast split segmentation'*
- *Translate the strategy – use algorithm 'contrast split segmentation'*
- *Review result – medium and steep objects are separated*

Next step: Develop strategy for next analysis step!



Lesson 5 Classifying steep areas

This lesson covers the following content

- *Strategy to classify objects with steep slope*
- *Translating the strategy – use 'Mean of dsm_slope'*
- *Review results – steep areas are classified*

5.1 Strategy to classify objects with steep slope

Introduction

An additional image layer was created, which contains information about slope, and objects were created representing the areas off different slope.

In a first classification step now the **steep areas** must be defined. They are later basis for creating the building objects.

Trees and buildings are both objects with steep slopes, which could lead to misclassifications. This means, they later on they must be excluded from further analysis to avoid misclassifications.

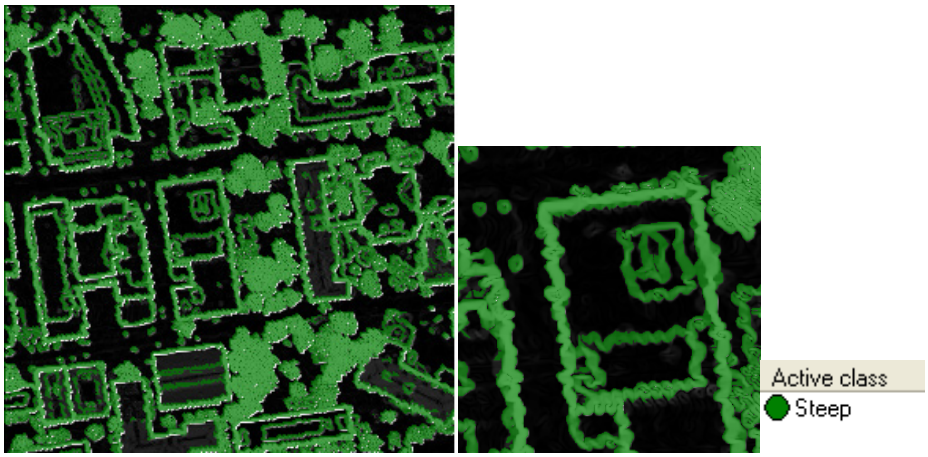
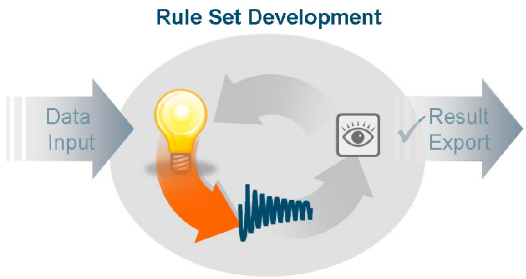


Figure 35: The steep areas are classified.

Next step: Translate Strategy into Rule Set!



5.2 Translating the strategy – use 'Mean of dsm_slope'

This chapter covers the following content

- ➔ *Find a threshold to classify steep areas*
- ➔ *Insert process to classify steep objects*

5.2.1 Find a threshold to classify steep areas

Information

The created layer 'dsm_slope' can be used as any other image layer. Using the mean DSM values steep from not steep areas can be separated.

Action!

1. In the 'Feature View' window browse to '**Object features>Layer Values>Mean**'.
2. Double-click on the feature '**dsm_slope**' and '**Update Range**' from the context menu.
3. **Increase the lower range** to the value 50. All relevant steep areas have slope values above 50.

Result Check

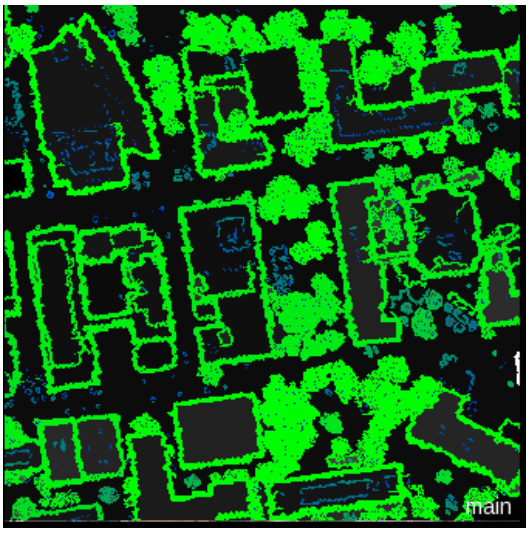


Figure 36: Feature View with values more than 50.

5.2.2 Insert process to classify steep objects

Action!

1. Collapse the parent process 'Segmentation', right-click on it and select 'Append New'. Name the process '**Classification**'.
2. Right-click on the new parent process and select 'Insert Child' from the context menu and name the process '**Classification Steep Areas**'.
3. Right-click on the last process and select 'Insert Child' from the context menu.
4. Select 'assign class' form the algorithms list.

5. Insert the condition '**Mean dsm_slope**' **more than 50**.
6. In the 'Use class' field enter the name '**Steep**' and choose green as color. Confirm with OK.
7. **Execute** the process.

Settings Check

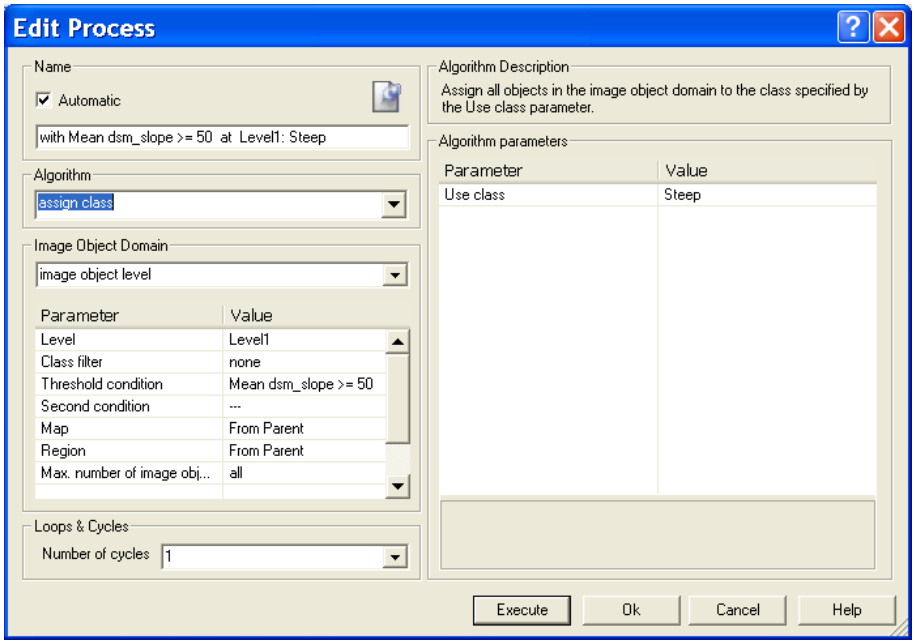


Figure 37: Process settings for assigning objects to the class 'Steep' with a slope equal or bigger than 50.

Rule Set Check

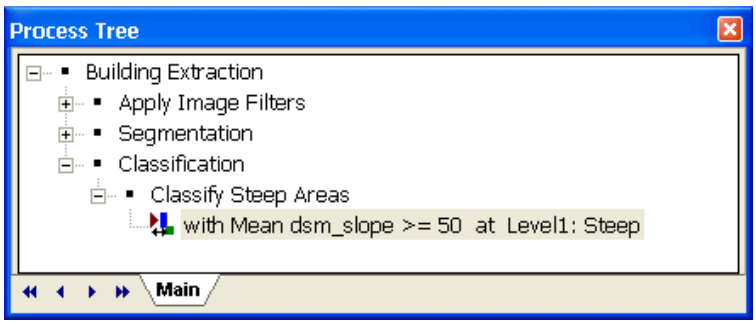
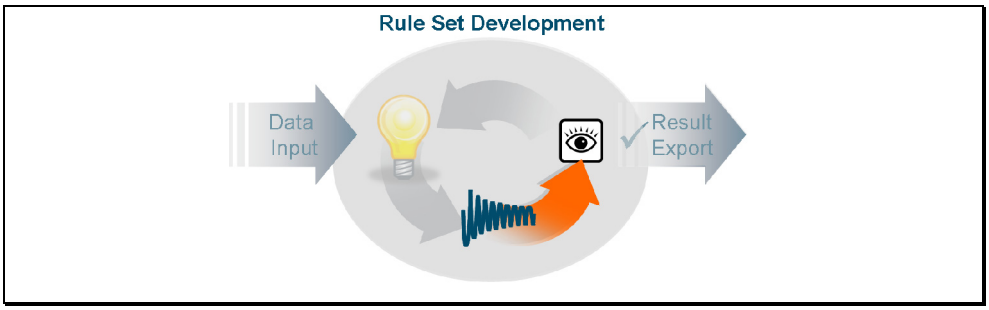


Figure 38: Process Tree with classification process added.



5.3 Review results – steep areas are classified

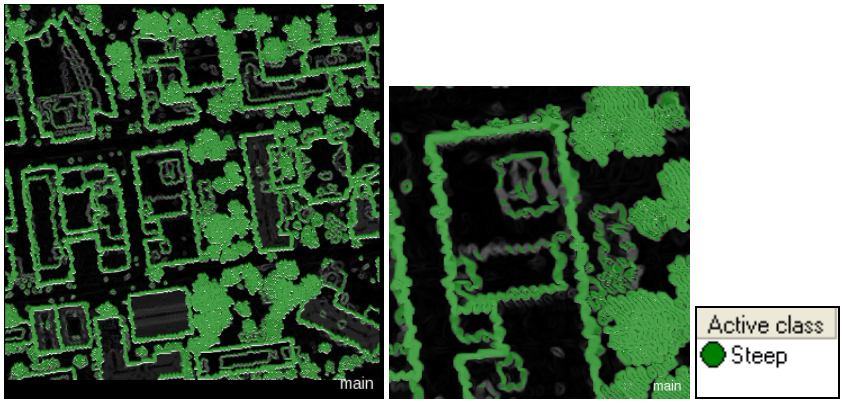


Figure 39: The steep areas are classified.

Result Check

The 'Steep' objects are classified. With the effect that there is a border of 'Steep' objects around the buildings.

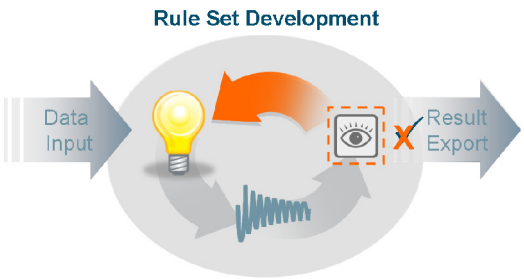
As trees are now classified as 'steep' too and the buildings will be classified only from 'unclassified' objects. Misclassifications are avoided in the further analysis.

The next step will be to classify the non elevated objects.

Lesson 5 covered the following content

- *Strategy to classify objects with steep slope*
- *Translating the strategy – use 'Mean of dsm_slope'*
- *Review results – steep areas are classified*

Next step: Develop strategy for next analysis step!



Lesson 6 Classifying the ground

This lesson covers the following content

- *Strategy to classify the ground – measure the lowest elevation*
- *Translating the first step of strategy – compute a variable*
- *Translating the second step of strategy – use 'Min. pixel value'*
- *Review result*

6.1 Strategy to classify the ground – measure the lowest elevation

Introduction

Aim of this Guided Tour is to classify using auto adaptive and/or stabile rules if necessary.

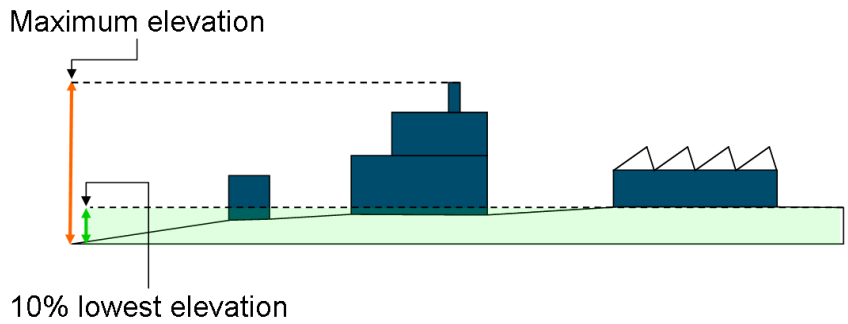
- The objects have been created using the 'contrast split segmentation', which calculates the threshold for splitting for every scene new.
- To classify steep areas a fixed threshold was used, because the borders of buildings are always very steep, therefore the steepness is a stabile information.

The next step in this analysis will be to **classify the ground**, respectively the non elevated area.

If you imagine how you recognize buildings in a bigger scene, it is obvious that you zoom into the scene and then compare the **elevation of a building in contrast to its neighbored ground.**

To be able to compare elevated objects to non-elevated, first objects which are for sure not elevated must be classified.

The assumption can be made, that in a certain area the terrain is not ascending more than a certain percentage, and use this assumption as a basis for rules to classify ground.



This means, those objects as ground have to be classified, which are **not above a certain elevation percentage** compared to the maximum elevation.

In our case, as the terrain is quite flat anyway, **10% of the maximum** elevation work fine.

One way to classify the objects with the lowest elevation is to use the mean value on the objects.

Introduction

A better solution to assess the elevation of objects is to refer to the minimum pixel value contained in an object. This ensures that only those objects are classified as ground, which have **all** pixels above the threshold.

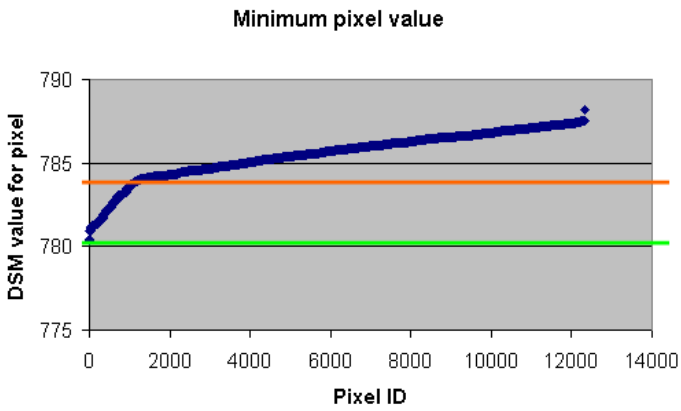


Figure 40: Green line represents the minimum pixel value of the objects, the orange line is the mean value.

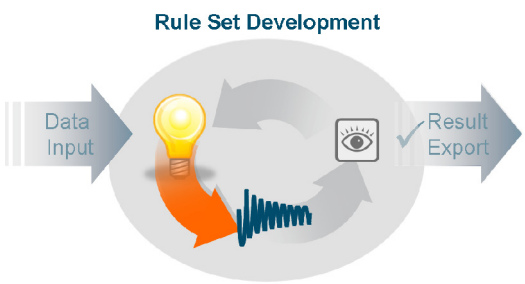
To achieve the classification based on a measurement of the 10% lowest elevated objects, two processing steps are needed:

- Measure the 10% lowest elevated objects and store the value in a so-called 'variable'
- Classify all objects with minimum pixel value lower than this variable as 'Ground'.

This makes the classification robust, because the value is re-calculated for every new scene.

The algorithm 'compute statistical value' performs a **statistical operation** on the feature distribution (here mean value for DSM) within an image object domain (here all unclassified objects) and stores the result in a **process variable**.

Next step: Translate Strategy into Rule Set!



6.2 Translating the first step of strategy – compute a variable

This chapter covers the following content

- ➔ *Statistical operation type 'quantile'*
- ➔ *Compute the 10th quantile of 'Mean DSM'*
- ➔ *Execute the process and check the measured value*

6.2.1 Statistical operation type 'quantile'

Information

The quantile returns the feature value, an object has at a specified percentage from all objects of the selected image object domain.

In our case it takes the 10% objects with the lowest pixel values for Layer DSM.

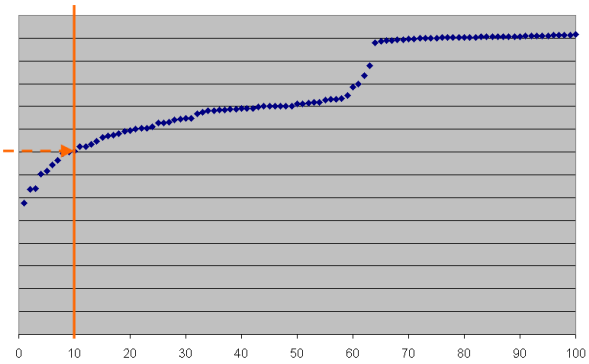


Figure 41: The orange line marks the 10th object of 100 with the lowest DSM value.

6.2.2 Compute the 10th quantile of 'Mean DSM'

Preparation

Action!

1. Collapse the parent process 'Classification', right-click on it and select 'Append New'. Name the process '**Classify Ground**'.
2. Insert a child process and name it '**Measure lowest elevation'**.

Add the process

1. Insert a child process and choose '**compute statistical value'** as algorithm.
2. Set '**unclassified**' as 'Class filter'.

Define the variable in which the computed value is stored.

3. Enter the name **'th_ground'** in the '**Variable**' field. Herewith you define the name of the variable.

4. Click anywhere **outside** the field.

The 'Create Scene Variable' dialog box opens.

5. Confirm the default settings in the 'Create Scene Variable 'dialog box with **OK**.

The variable '**th_ground**' is created.

Define the operation mode and the feature to be measured

6. From the '**Operation'** dropdown menu choose '**quantile'**.

7. In the 'Parameter' field insert the value **10**.

8. In the **'Feature**' field define the feature '**Mean DSM**'.

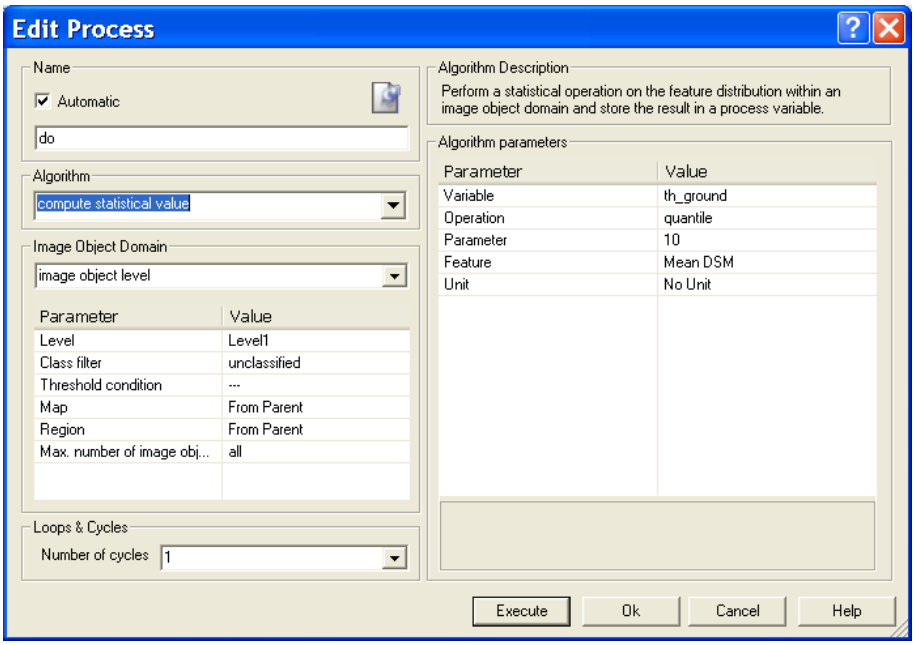


Figure 42: Process settings for computing the 10th quantile for the feature 'Mean DSM'.

Settings Check

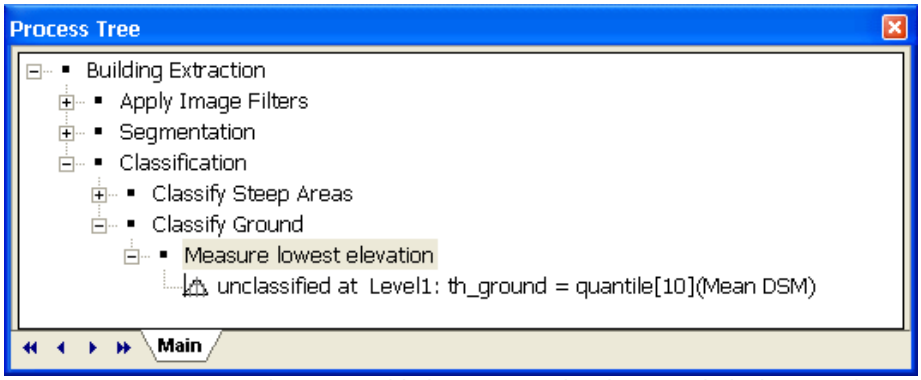


Figure 43: Process Tree with process added to measure the objects with the lowest value.

Rule Set Check

6.2.3 Execute the process and check the measured value

Action!

1. **Execute** the process.
2. Check the calculated value in the **'Image Object Information'**.

Result Check

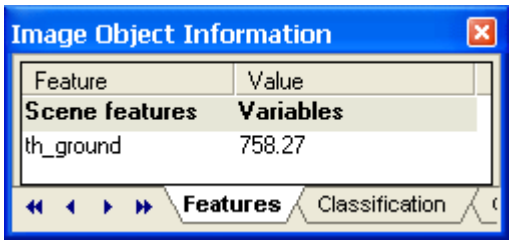


Figure 44: The calculated variable in the 'Image Object Information' window.

6.3 Translating the second step of strategy – use 'Min. pixel value'

This chapter covers the following content

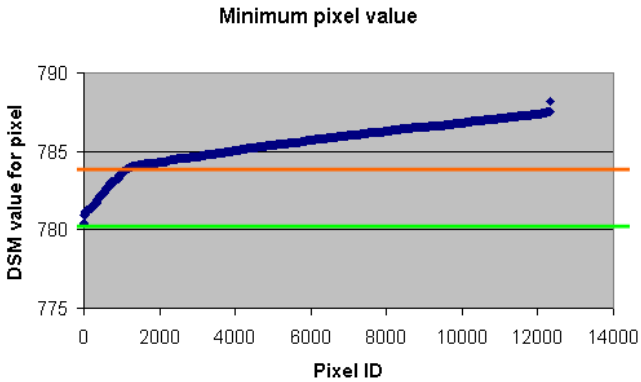
→ *Check the automatically calculated threshold for feature 'Min. pixel value'*

→ *Classify using the variable*

6.3.1 Check the automatically calculated threshold for feature 'Min. pixel value'

Information

As discussed in the 'Strategy' section feature giving back the lowest pixel value of an object is needed.



1. In the 'Feature View' window browse to '**Object features>Layer Values>Pixel-based**'.
2. Double-click on 'Create new 'Min. pixel value'' and select DSM.
3. Decrease the maximum values to 758, which is about the measured 'th_ground' variable value.

Action!

Figure 45: The Feature View of 'Min. pixel value of DSM'.

Result Check

All objects in the range are ground objects.

6.3.2 Classify using the variable

4. Append a new process in parallel to 'Measure lowest statistics' and choose algorithm 'assign class'.
5. As domain define 'unclassified'.
6. As condition set the feature 'Min. pixel value of DSM' **lower than the variable 'th_ground'**.
7. Click in the 'Value' field and select '<Create new class>'. Enter the name 'Ground' and choose dark blue as color. Confirm with OK.
8. Execute the process.

Action!

Settings Check

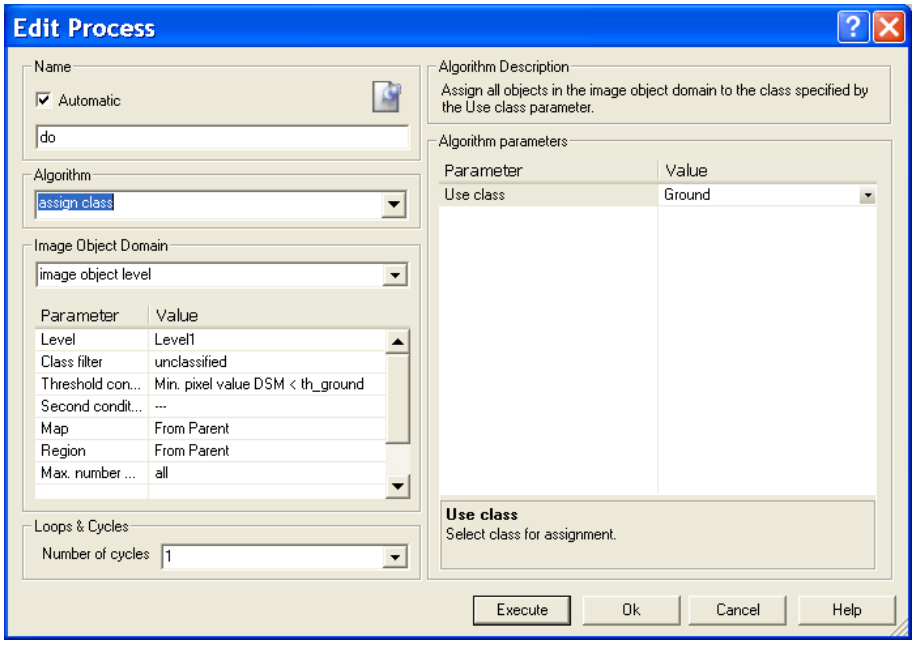


Figure 46: Process settings for classifying 'Ground' objects.

Rule Set Check

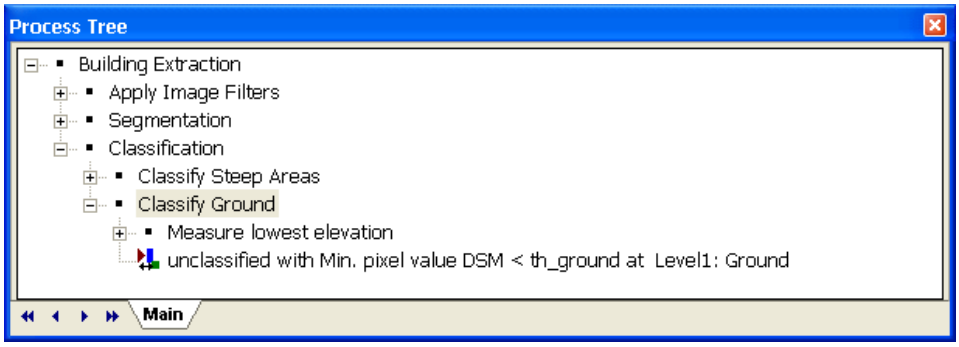
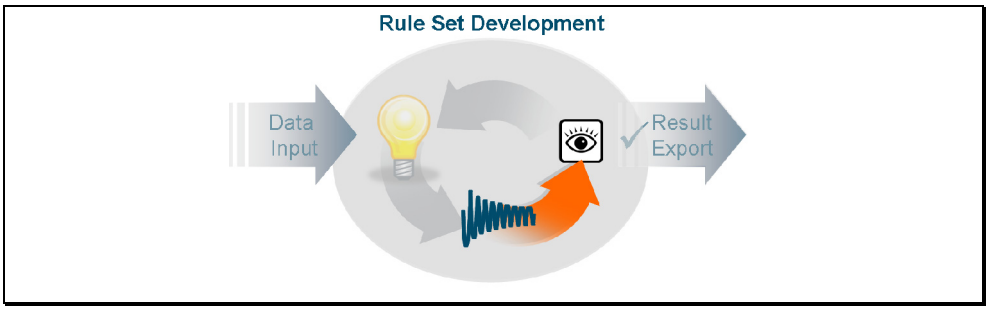


Figure 47: Process Tree with classification process added.



6.4 Review result – ground is classified

All unclassified objects with a lower value than measured are now classified as 'Ground'. Most of the remaining unclassified objects are buildings.

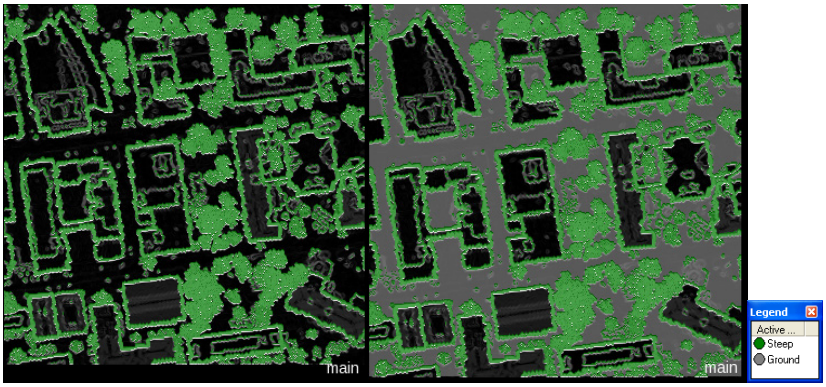


Figure 48: Left: 'Steep' objects are classified; Right: 'Ground' objects are classified.

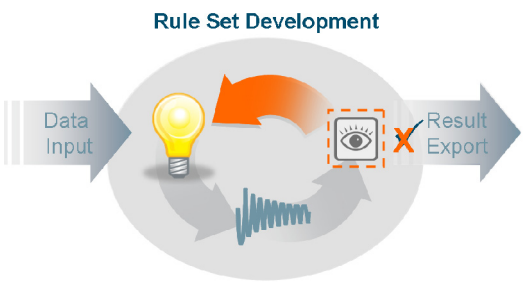
Result Check

In the next step the elevation of the remaining unclassified objects compared to the 'Ground' objects is computed.

Lesson 6 covered the following content

- → *Strategy to classify the ground – measure the lowest elevation*
- → *Translating the first step of strategy – compute a variable*
- → *Translating the second step of strategy – use 'Min. pixel value'*
- → *Review result*

Next step: Develop strategy for next analysis step!



Lesson 7 Classifying buildings

This lesson covers the following content

- *Strategy to classify buildings - the difference in elevation*
- *Translating the strategy – use difference in elevation*
- *Review result – basic building shapes are classified*

7.1 Strategy to classify buildings - the difference in elevation

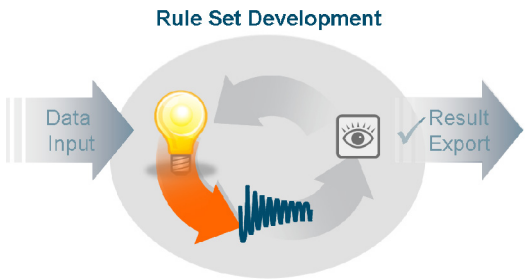
Introduction

Buildings have a significantly different elevation than its neighborhood. In opposition to a fixed height value the **difference in elevation** is a stabile and transferable characteristic of buildings.

In rather flat areas, or if you are analyzing only areas of a certain size (tiling and stitching approach), you can compare the elevation of the individual buildings to the average ground elevation of the subset.

This means you have to bring the elevation of the individual buildings in relation to the elevation of the ground.

Next step: Translate Strategy into Rule Set!



7.2 Translating the strategy – use difference in elevation

This chapter covers the following content

- ➔ *Create 'Customized Relational Feature'*
- ➔ *Find threshold for 'Building' classification*
- ➔ *Insert process using the created feature and the algorithm 'assign class'*

Introduction

There are several features available within eCognition Developer that express differences in feature values. The unclassified objects must be compared with the 'Ground' objects, respectively to a specific class. These kind of features can be found in the feature category 'Class-Related'.

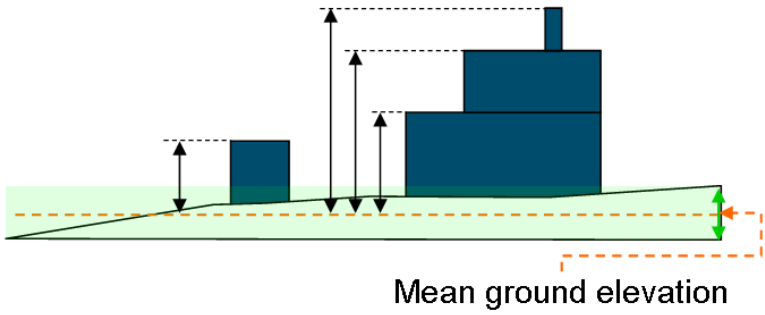
Per default there is a 'Mean difference to neighbors' feature, but this feature compares only directly adjacent objects.

The potential buildings are all surrounded by 'Steep' objects, therefore this feature does not fit for analysis, the unclassified objects would be 'undefined', because they are too far away.

Similar to the 'Customized Arithmetic Features' you can also create 'Customized Relational Features' to your needs.

We need a feature expressing:

Calculate the level mean difference to 'Ground' objects for Layer DSM.



7.2.1 Create 'Customized Relational Feature'

1. To open the 'Edit Customized Feature' dialog box, select either:
   - '**Tools > Manage Customize Features> Add**'
   - or **double-click** in the **Feature View** under '**Object features > Customized'** on the '**Create new 'Relational Feature'**'.

Action!

The customized feature shall express:

Compare the mean difference of an individual image object with the Mean of DSM of all 'Ground' objects of a level.

Settings Check

Action!

2. In the '**Feature name**' line enter the name ❶ '**MeanDiff_MeanDSM_Level_Ground**'.
3. Select ❷ '**Mean difference**' from the drop down menu in the 'Rational function' section.
4. In the '**Feature Selection**' select: 'Object features >Layer Values>**Mean>DSM**'❸.
5. In the '**Relational function**' section select ❹ '**level**', the feature will refer to all objects of the level.
6. As '**Level Distance**' keep 0 .This defines that the evaluation is done in the same level, not one above (1) or below (-1).
7. In the '**Class Selection**' ❻ select: '**Ground**'. This specifies the class to refer to.
8. Click '**OK**' to create the new feature.

Settings Check

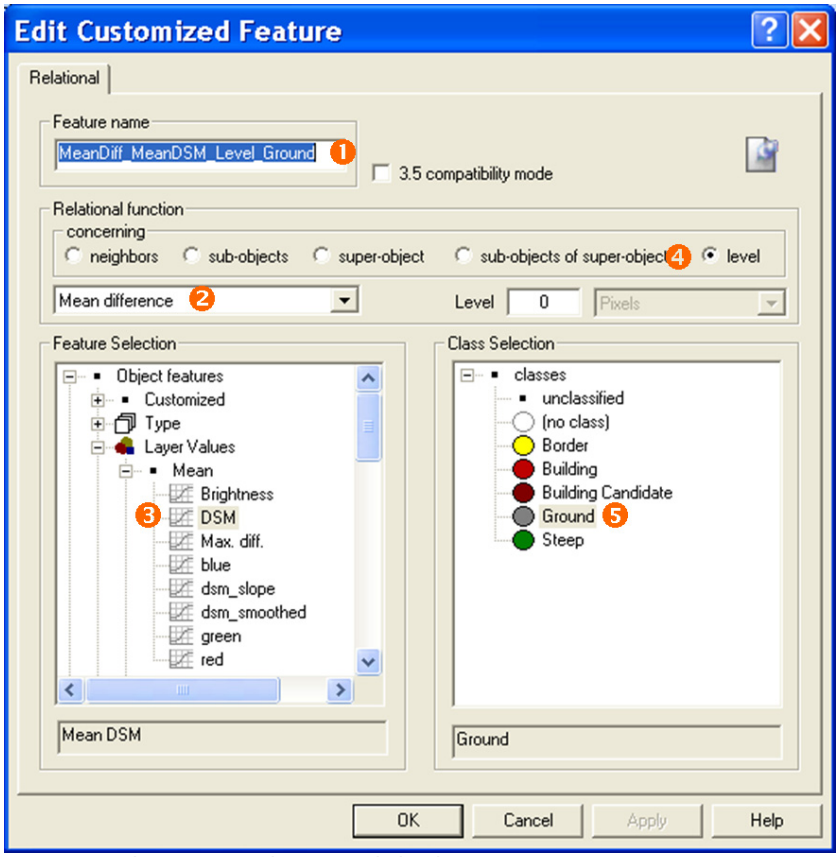


Figure 49: 'Edit Customized Feature' dialog box.

Check Feature View

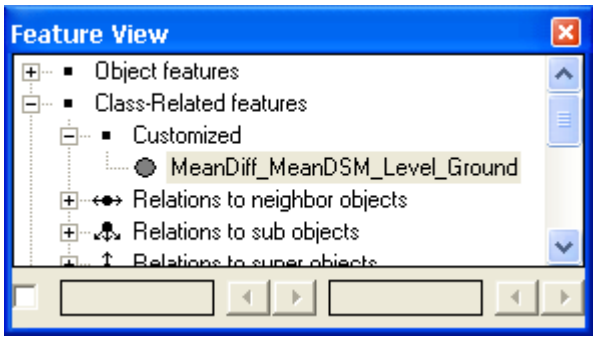


Figure 50: The created feature is listed under 'Class-Related features>Customized'.

7.2.2 Find threshold for 'Building' classification

Information

Negative values for this feature express that the object has a lower value than the average Mean DSM for 'Ground objects, positive values, that they have a higher value. With a mean difference of more than10 to the average elevation of the level most buildings are represented.

Action!

1. In the 'Feature View' window browse to '**Class-Related features>Customized**'.
2. Double-click on the feature '**MeanDiff_MeanDSM_Level_Ground**', right-click and '**Update Range**' from the context menu.
3. Increase the lower values until you reach 10.

Result Check

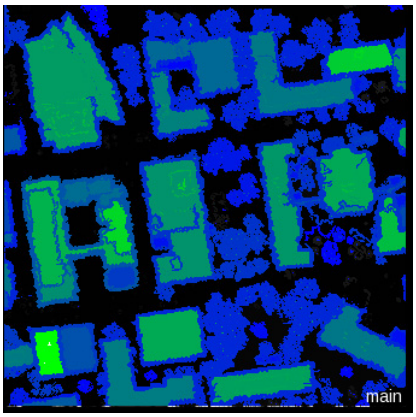


Figure 51: Feature View of value range 10 to 100 for the feature 'MeanDiff_MeanDSM_Level_Ground'.

7.2.3 Insert process using the created feature and the algorithm 'assign class'

Action!

1. Collapse the parent process '**Classify Ground**', right-click on it and select 'Append New'. Name the process '**Classify Buildings**'.
2. Insert a child process and choose '**assign class'** as algorithm.
3. Set the 'Class filter' to 'unclassified'.
4. Insert as 'Threshold condition': '**MeanDiff_MeanDSM_Level_Ground**' equal or larger than **10**.
5. Create the class '**Building**' and choose red as color. Confirm with OK.
6. **Execute** the process.

All 'unclassified' objects with an equal or higher value of 10 for the customized feature are classified as 'Building'.

Settings Check

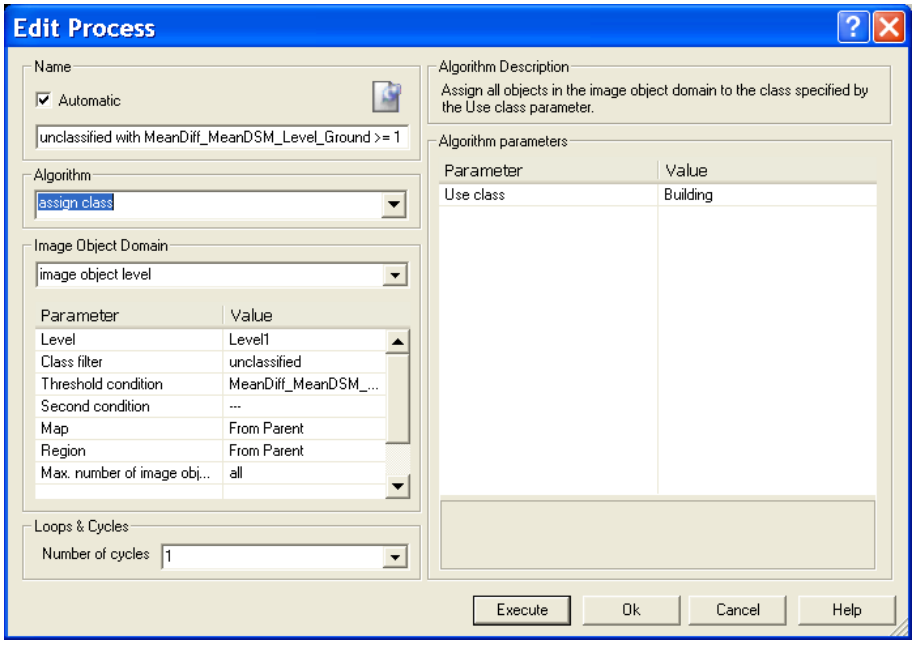


Figure 52: Process settings for classifying 'Building' objects using the customized Class-Related feature.

Rule Set Check

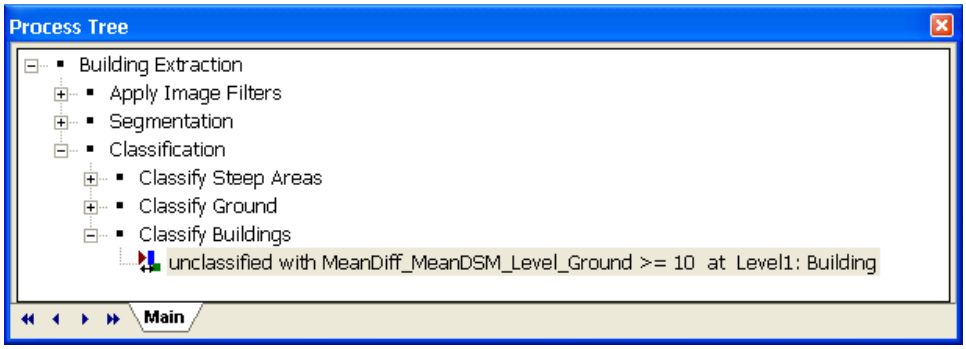
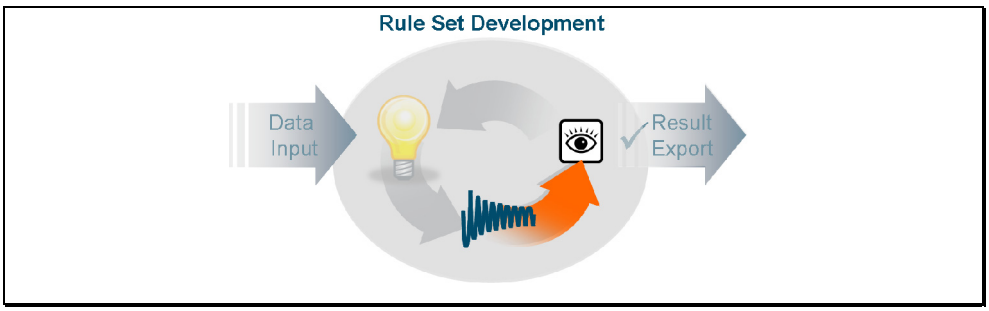


Figure 53: Process Tree with classification process added.



7.3 Review result – basic building shapes are classified

The initial 'Building' objects are classified. But there are misclassifications and some 'Steep' objects actually belong to the class 'Building'. A refinement is necessary.

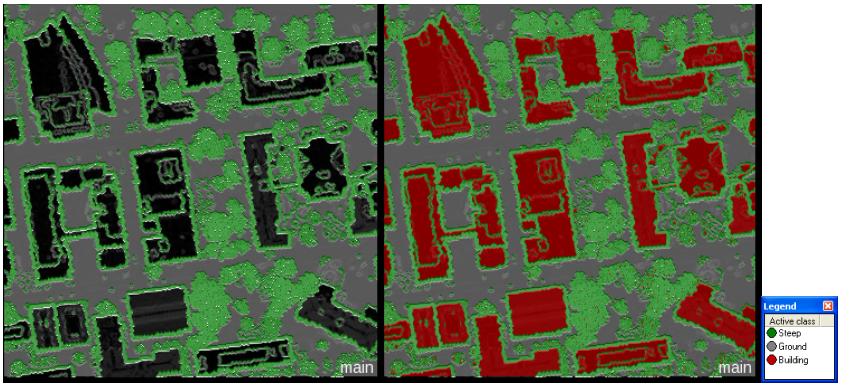


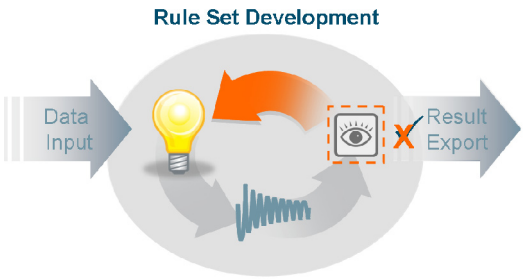
Figure 54: Left: Classification view before 'Building' classification; Right: Classification view after 'Building' classification.

Result Check

Lesson 7 covered the following content

- *Strategy to classify buildings - the difference in elevation*
- *Translating the strategy – use difference in elevation*
- *Review result – basic building shapes are classified*

Next step: Develop strategy for next analysis step!



Lesson 8 Refining the 'Building' classification

This lesson covers the following content

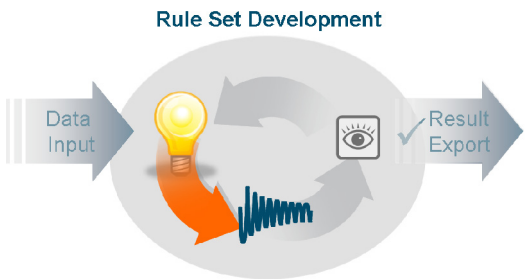
- *Strategy to refine 'Building' classification - 'Area' feature*
- *Translating strategy – use 'Area' feature for refinement*
- *Review result – too small 'Building' objects are declassified*

8.1 Strategy to refine 'Building' classification - 'Area' feature

Introduction

Buildings have a certain size, this characteristic will be used to de-classify misclassified objects which are too small.

Next step: Translate Strategy into Rule Set!



8.2 Translating strategy – use 'Area' feature for refinement

This chapter covers the following content

- ➔ *Find a threshold for de-classifying too small 'Building' objects*
- ➔ *Insert process to refine classification by using the 'Area' feature*

8.2.1 Find a threshold for de-classifying too small 'Building' objects

Action!

1. In the 'Feature View' window browse to '**Object features>Geometry>Extent**'.
2. If the unit of the feature 'Area' is not set to 'pixels', right-click on it and select 'Edit unit' from the context menu. Select the unit '**Pixels**' from the drop-down menu.
3. Double-click on the feature '**Area**', right-click and select '**Update Range**' from the context menu.
4. Increase the lower values until you reach 2000.

Objects smaller than 2000 pixel are not displayed in the color range.

8.2.2 Insert process to refine classification by using the 'Area' feature

1. Append a new process and choose algorithm 'assign class'.
2. As domain define 'Building'.
3. As condition set the Feature 'Area' **lower than 2000**.
4. Click in the 'Value' field and select '**Steep**'. Confirm with OK.
5. **Execute** the process.

All 'Building objects' smaller than 2000 pixel will be classified as 'Steep'.

Settings Check

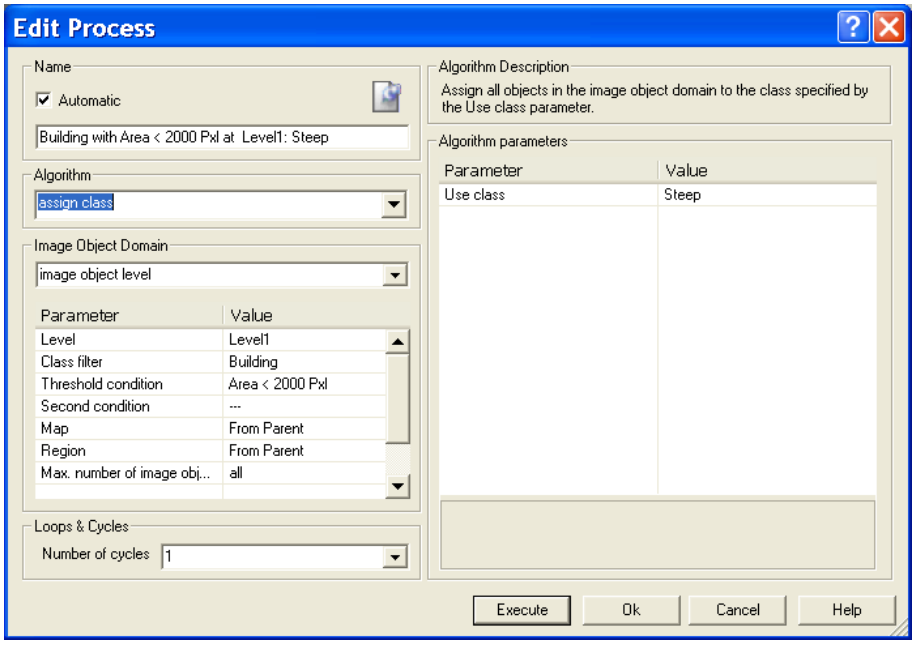


Figure 55: Process settings for refine classification of 'Building' objects.

Rule Set Check

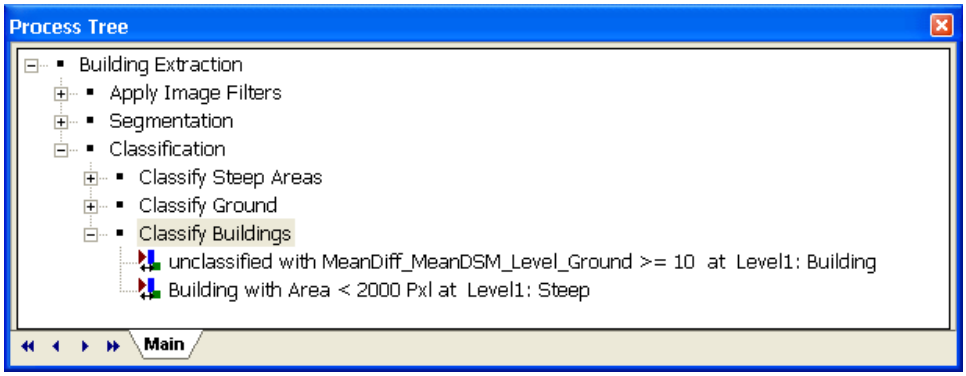
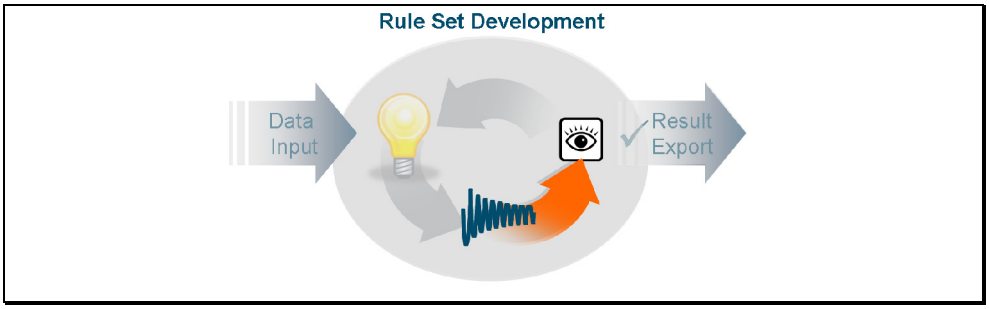


Figure 56: Process Tree with classification process added.



8.3 Review result – too small 'Building' objects are declassified

The too small 'Building' objects are now declassified, but there are 'Steep' objects within and around the 'Buildings' which actually belong to it.

To get closer to the optimal outlined object, the 'Building' objects have to grow into the surrounding 'Steep objects.

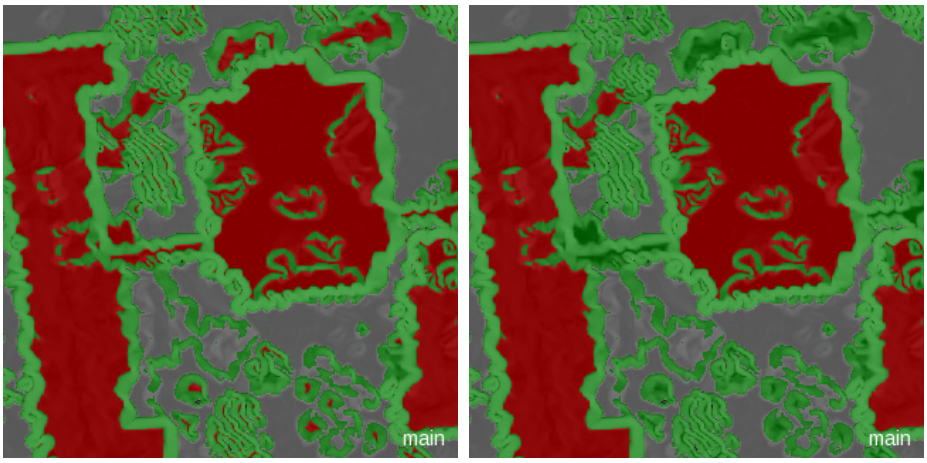


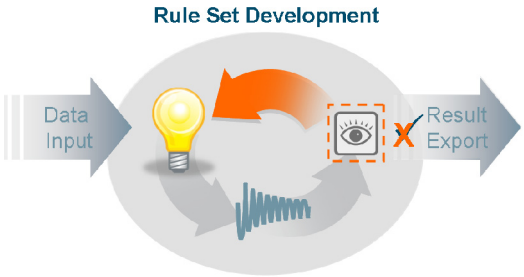
Figure 57: The too small 'Building' objects are now classified as 'Steep'

Result Check

Lesson 8 covered the following content

- *Strategy to refine 'Building' classification - 'Area' feature*
- *Translating strategy – use 'Area' feature for refinement*
- *Review result – too small 'Building' objects are declassified*

Next step: Develop strategy for next analysis step!



Lesson 9 Cleaning up 'Steep' objects

This lesson covers the following content

- *Strategy to reshape objects*
- *Translating strategy - reshape using the spectral layers*
- *Review result – objects are reshaped*
- *Strategy to grow into spectrally similar objects*
- *Translating strategy – classify adjacent, spectrally similar objects*
- *Review the result – spectrally similar objects are classified*
- *Strategy to fill holes in 'Building' classification*
- *Translating strategy – find surrounded and enclosed*
- *Review result – enclosed objects are classified*

Introduction

This lesson to clean up the 'Steep' objects has three major parts.

- Reshape the 'Steep' objects to smaller ones using spectral image layers.

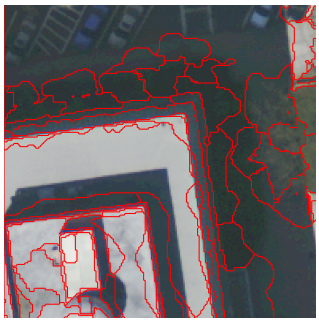


Figure 58: Detail: Outlines after reshaping of 'Steep' objects.

- Grow the 'Building' objects into spectrally similar neighbors

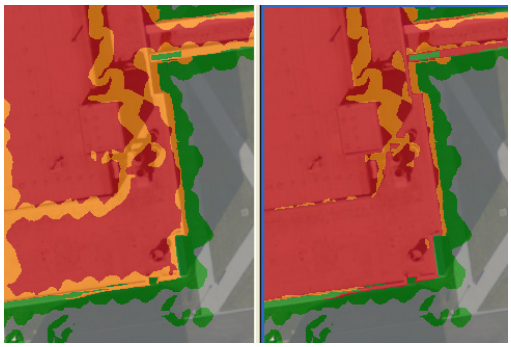


Figure 59: Before and after the spectral similarity classification.

Introduction

- Fill holes in the 'Building' classification

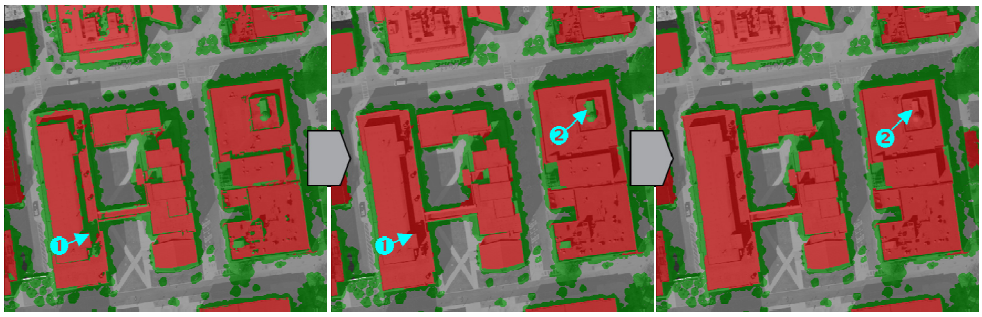


Figure 60: Left: Before first processing step, Middle: after first processing step; Right: After second processing step.

9.1 Strategy to reshape objects

Introduction

Around and within the buildings there are objects classified as 'Steep'.

To classify the complete building the initial 'Building' objects have to be expanded into 'Steep' objects.

A couple of processing steps are needed to check whether these objects belong to the class 'Building' or not.

The current objects were created on basis of the slope layer. To analyze whether some 'Steep' objects belong to the class 'Building' or not, first they must be reshaped into smaller objects. To be able to use also spectral information for classification, such as spectral similarity, these new objects must be **created on basis of the spectral layers**.

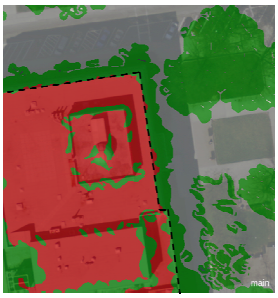
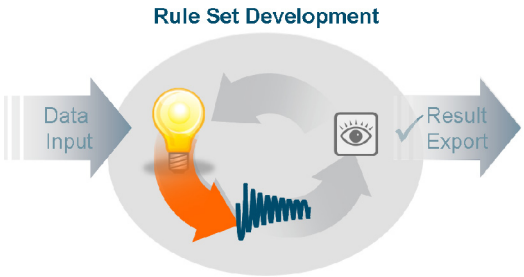


Figure 61: Ideal outline for a building, some parts are currently classified as 'Steep'.

Next step: Translate Strategy into Rule Set!



9.2 Translating strategy - reshape using the spectral layers

This chapter covers the following content

➔ *Prepare process structure*

➔ *Insert the 'multiresolution segmentation' process as a child process*

Information

The 'multiresolution segmentation' is used to reshape the 'Steep' objects, respectively to create smaller objects based on the spectral layers. Therefore the algorithm must point to the already existing '**Steep**' objects and the setting '**Use current**' must be defined to split up the objects.

9.2.1 Prepare process structure

Action!

1. Collapse the parent process 'Classify Buildings', right-click on it and select 'Append New'. Name the process '**Clean Up 'Steep'**'.

9.2.2 Insert the 'multiresolution segmentation' process as a child process

2. Select the inserted process and **right-click** on it. Choose '**Insert Child**' form the context menu.
3. In the '**Algorithm**' field choose '**multiresolution segmentation**' from the list.

Define the Image Object Domain

4. Change the Image Object Domain from 'pixel level' (1) to '**image object level**'.
5. Change the 'Class filter to '**Steep**' (2).
6. In the field 'Level Usage' change from 'use current merge only' to '**Use current**' (3).

This indicates that the objects will be split up. If you keep 'use current merge only' small objects are merged together.

Define the segmentation parameters

7. Expand the field 'Image layer weights' (4) and set:
   - Dsm_smoothed, dsm_slope and DSM to 0.
   - Red, green and blue to 1.

This defines that only the red, green and blue layer will be used for segmentation.

8. In the field 'Scale parameter' (4) enter 25.

9. Keep all other default settings.

10. Execute the process.

Settings Check

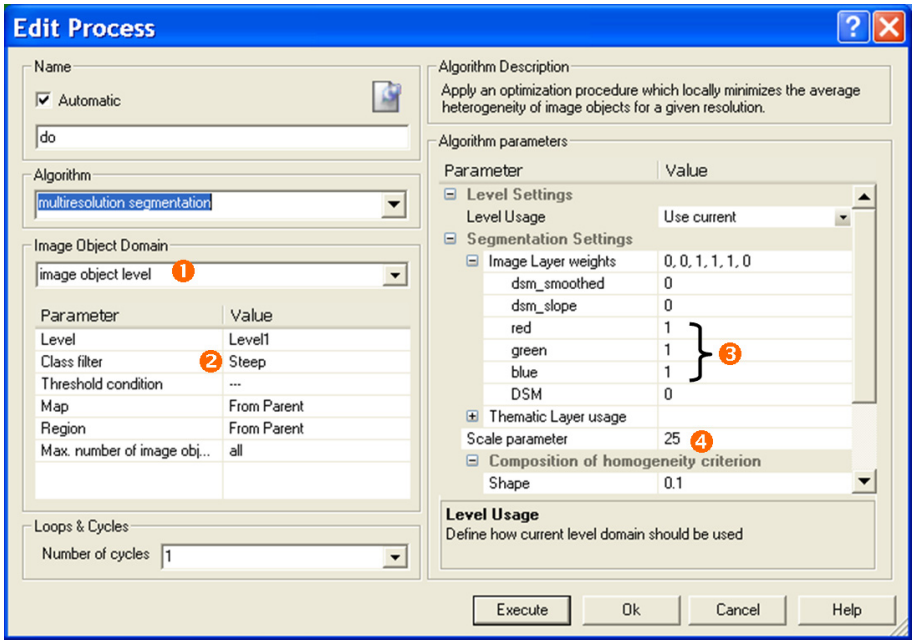


Figure 62: 'Edit Process' dialog box with settings to reshape 'Steep' objects in 'Level 1' using the multiresolution segmentation algorithm.

Rule Set Check

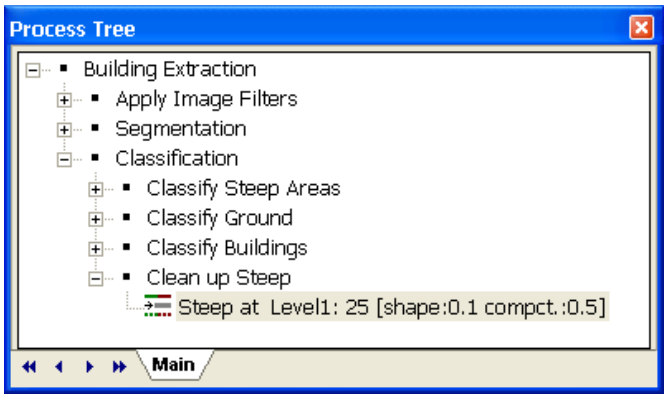
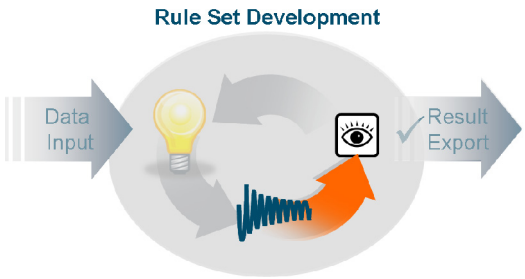


Figure 63: Process Tree with segmentation process added.

Next step: Review Result!



9.3 Review result – objects are reshaped

The borders of the buildings are now finer segmented and ready for further classification. Now the basis is created to grow the initial 'Building' objects into the adjacent 'Steep'.

Result Check

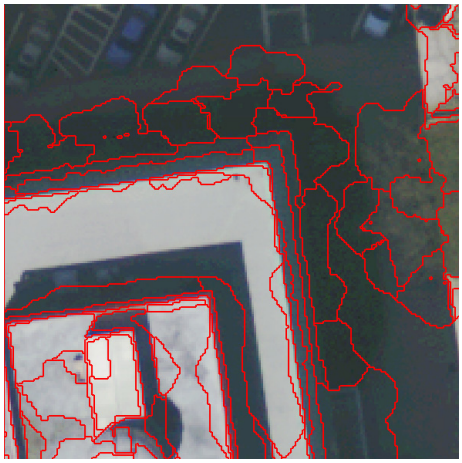
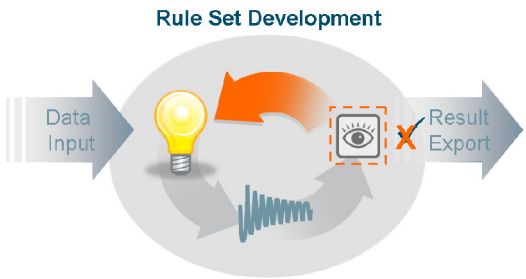


Figure 64: Detail: Outlines after reshaping of 'Steep' objects.

Next step: Develop strategy for next analysis step!



9.4 Strategy to grow into spectrally similar objects

This chapter covers the following content

- ➔ *Concept of 'temporary classes'*
- ➔ *Spectral similarity features*

Introduction

Some of the objects now classified as 'Steep' are spectrally very similar to neighboring 'Building' objects.

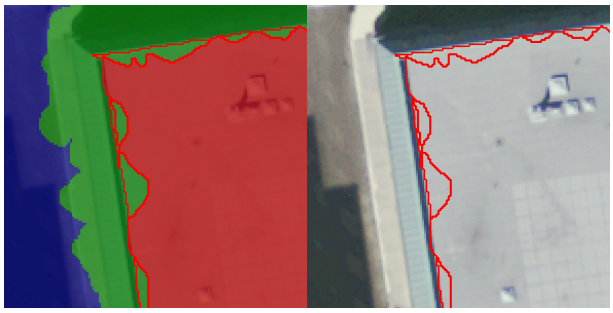


Figure 65: Selected objects (red outline) which are spectrally similar to the 'Building' object, but not yet classified as 'Building'.

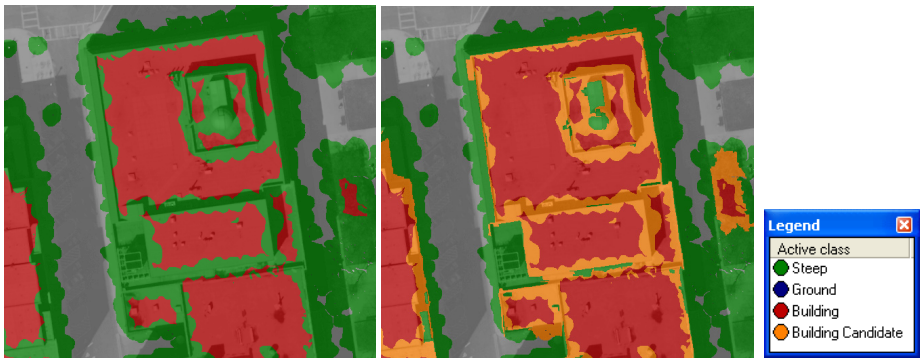
9.4.1 Concept of 'temporary classes'

Introduction

To avoid misclassifications, you can restrict an analysis to a certain group of objects using a temporary class, which will be declassified after the analysis step.

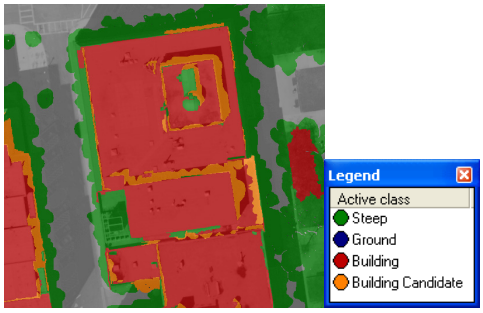
In the current example the analysis is restricted to objects directly adjacent to 'Building' objects, represented by a separate class 'Building Candidate'.

- Those 'Steep' objects directly adjacent to 'Buildings' are classified as a temporary class 'Building Candidates'.



Introduction

- Then only the 'Building Candidates' are analyzed according to their spectral similarity



- In the end the remaining 'Building Candidates' are classified as 'Steep' again.

9.4.2 Spectral similarity features

Introduction

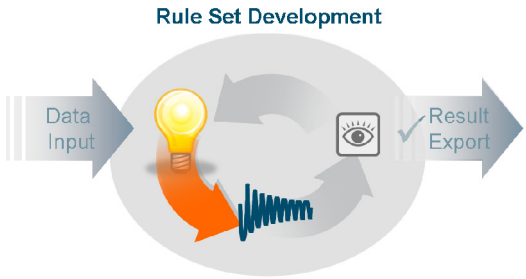
The spectral similarity of the adjacent objects for classification needs to be analyzed. As roofs have a high variety in color the spectral similarity for all channels, red, green and blue has to be evaluated at once.

The easiest way to do this is to refer to the brightness of an object. The brightness must be defined as (red+green+blue)/3.

But this is not enough, the brightness has to be linked to the neighboring 'Building' objects. This can be expressed by comparing the differences in brightness to neighboring 'Building' objects.

Similar to evaluating the difference in elevation, the difference in spectral similarity to 'Building' objects is evaluated by creating a customized Class-Related feature. It is important, that the feature is not differentiating between negative similarity and positive similarity. Respectively a little brighter or a little darker. The 'little' is important.

Next step: Translate Strategy into Rule Set!



9.5 Translating strategy – classify adjacent, spectrally similar objects

This chapter covers the following content

- ➔ *Classify 'Steep' objects directly adjacent to 'Building' objects' as 'Building Candidates'*
- ➔ *Create the feature 'Customized Brightness'*
- ➔ *Excursus: Distance calculation in eCognition*
- ➔ *Set the distance calculation mode in the 'set rule set options' algorithm*
- ➔ *Create the customized relational feature expressing spectral differences/similarities*
- ➔ *Find a threshold to classify spectrally similar 'Building Candidates' to the class 'Buildings'*
- ➔ *Classify using the Customized Class-Related feature*
- ➔ *Classify the temporary class 'Building Candidates' back to the class 'Steep'*

9.5.1 Classify 'Steep' objects directly adjacent to 'Building' objects' as 'Building Candidates'

Find a feature and a threshold

Information

Within the 'Class-Related features' there are several features available expressing whether an object is adjacent or not. The feature '**Relative border to Buildings**' is used. The 'Relative border' must be more than 0 to express that an object is directly adjacent to a 'Building' object.

Classify 'Building Candidate' objects

Action!

1. Right-click on the segmentation process and select 'Append New'. Name the process '**Clean up spectrally similar neighbors**'.
2. Right-click on the parent process and select 'Insert Child'. Choose '**assign class'** as algorithm.
3. Change the 'Class filter' to 'Steep'.
4. Insert the feature 'Class-Related features>Relations to neighbors> **Rel. border to Building**'. Insert the condition **larger than 0**.
5. Create the class 'Building Candidate' and choose orange as color.
6. **Execute** the process.

All 'Steep' objects adjacent to 'Building' objects are now classified as 'Building Candidate'.

Rule Set Check

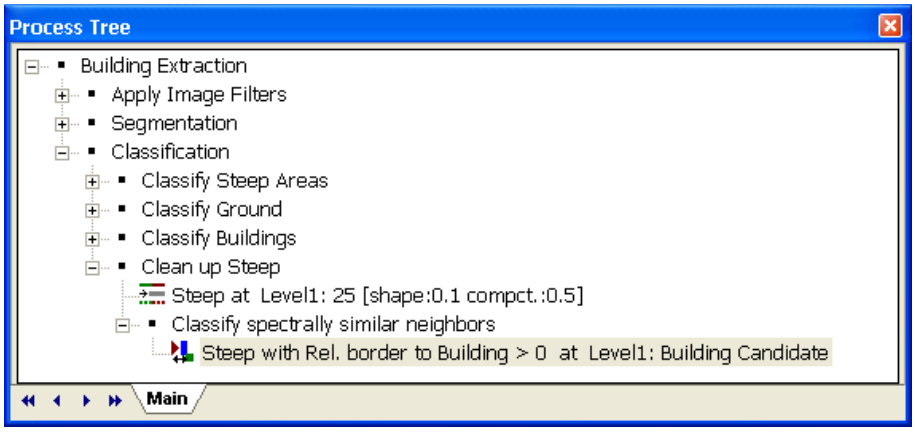


Figure 66: Process Tree with classification step added.

Review intermediate result

These 'Building Candidates' are now ready to be evaluated according to their spectral similarity.

Result Check

Figure 67: 'Steep' objects with border to 'Building' objects are classified as 'Building Candidate'.

9.5.2 Create the feature 'Customized Brightness'

Information

In the next processing step the spectral difference of objects is evaluated. The standard feature 'Brightness,' as it is per default listed in the Feature View, is always calculated on basis of **all image layers**. This means in our case also the DSM and the smoothed and slope layer are taken into account. This would falsify the result. A **customized feature** must be created, representing the **spectral brightness only**.

Action!

1. In the 'Feature View' window browse to 'Object Features>Customized' and double-click on '**Create new 'Arithmetic Feature''**.

The 'Edit Customized Feature' dialog box opens.

2. Select the features from the feature list 1 by double clicking, insert an operator by clicking on it in the operator field 2.

3. Confirm with 'OK'.

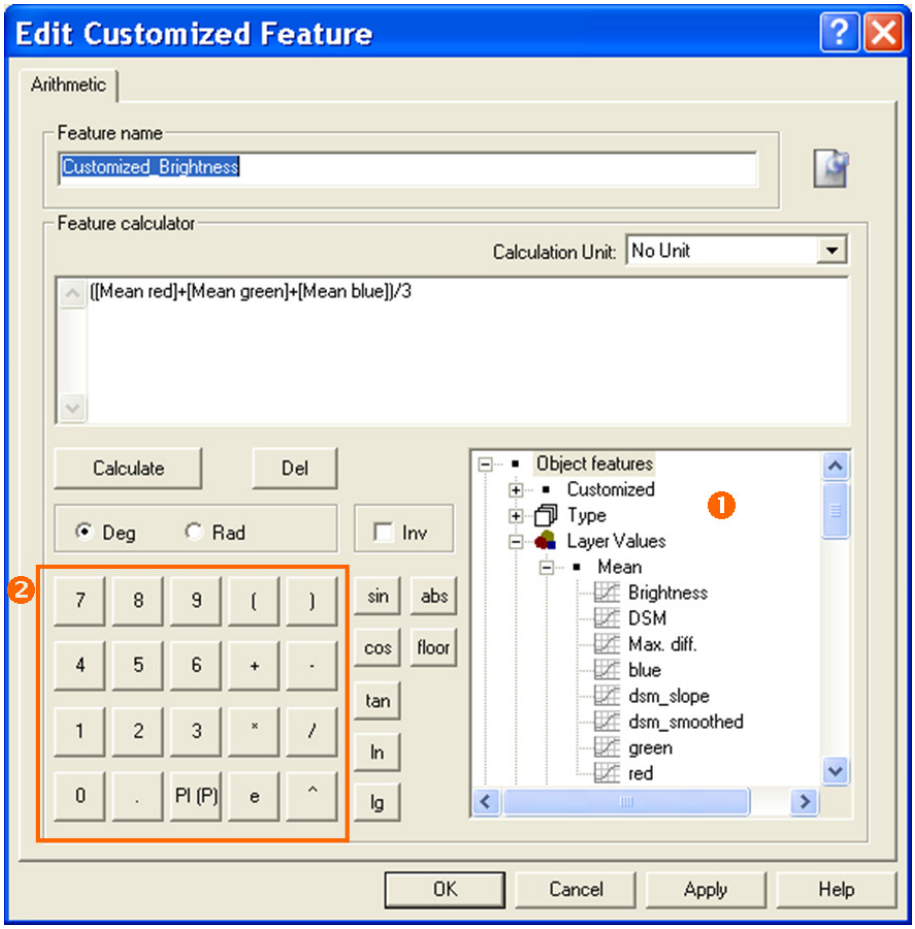


Figure 68: The 'Edit Customized Feature' dialog box with settings to create the feature 'Customized_Brightness'.

Settings Check

Note:

Alternatively right-click in the Feature View and select 'Load…' from the context menu. Browse to the folder …\GuidedTour\Data\Module2 and load the 'Cust_Brightness.duf'.

9.5.3 Excursus: Distance calculation in eCognition

Introduction

Before creating the customized relational feature to evaluate the spectral similarities of neighboring objects, the general distance calculation modes are explained. Depending on which mode is set, different values are calculated for distances.

With the customized relational features, there is the possibility to define the search range in which objects are evaluated according to their similarity. This means in this example not all objects are compared to a 'Building' object, but only those object **within a certain distance**. Here in our case a distance of **20 pixels** is sufficient.

eCognition offers two **methods** of calculating the distance of image objects which can be set with the **algorithm 'set rule set options'**.

- Distance calculation based on '**Smallest enclosing rectangle**'
- Distance calculation based on '**Center of gravity**'

A special situation is given if the objects are **direct neighbors**. With '**Smallest enclosing rectangle**' mode the distance is 0.

Within the 'Relations to neighbors' features there are several features using the distance calculation.

For the features '**Existence of**', '**Number of**' and '**Rel. area of**' the distance is set by the **user**. This means e.g. for the feature 'Existence of Water (10)', that in a search range of 10 pixel the existence of 'Water' objects is analyzed.

The feature '**Distance to**' calculates for every object the distance to the **closest** object of the specified class. For example, the feature 'Distance to Water' gives back the value for the closest 'Water' object.

Distance calculation methods 'Smallest enclosing rectangle' vs. 'Center of gravity'

The distance value can vary, due to whether the 'Smallest enclosing rectangle' or the 'Center of gravity' mode is set in the **'Set Rule Set Options' algorithm**.

'Center of gravity' mode

The center of gravity approximation measures the distance between the center of gravity between two image objects. This measure can be computed very efficiently but it can be quite **inaccurate** for **large image objects**.

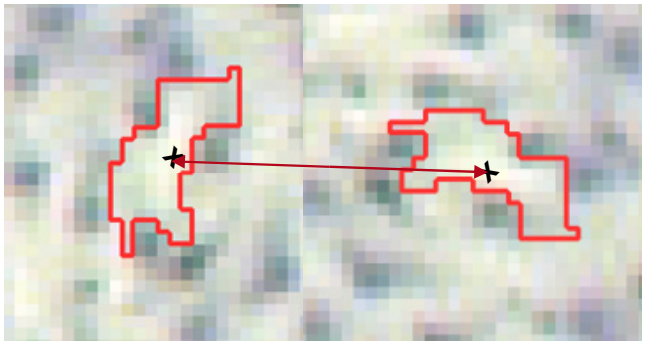


Figure 69: Example for distance calculation using 'center of gravity' mode.

'Smallest enclosing rectangle' mode

The smallest enclosing rectangle approximation tries to correct the center of gravity approximation by using rectangular approximations of the image object to adjust the basic measurement delivered by the center of gravity. The distance is measured from the **borders of the bounding boxes**.

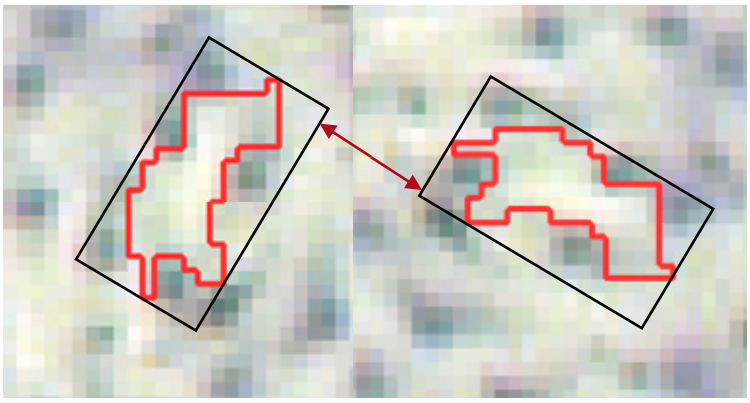


Figure 70: Example for distance calculation using 'smallest enclosing rectangle' mode.

The distance 0 for 'Existence of', 'Number of' and 'Rel. area of'

If the distance for the features '**Existence of**', '**Number of**' and '**Rel. area of**' is set to **0** then only the **direct neighbors** are considered. When the distance is **greater than 0** then the relation of the objects is computed using either their **centers of gravity or their smallest enclosing rectangle.**

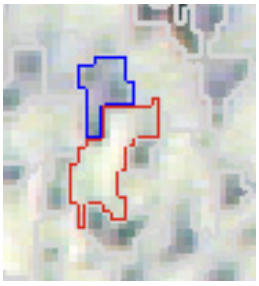


Figure 71: Example for distance 0.

The distance 0 for 'Distance of'

Adjacent objects have a distance of = 0 for 'smallest enclosing rectangle' mode. If 'center of gravity is selected, the distance of adjacent objects is always larger than 0.

With the 'set rule set options' algorithm, settings that control the behavior of the Rule Set can be defined. These settings were defined in the 'Tools>Options' menu in earlier versions of the software. Because these settings are part of the Rule Set now, they are preserved when the Rule Set is run on a server.

9.5.4 Set the distance calculation mode in the 'set rule set options' algorithm

Information

With this algorithm you can, besides the distance calculation, also define resampling methods, conditions for undefined feature values and polygon settings.

Action!

1. **Append** a new process and choose '**set rule set options**' as algorithm.
2. In the field '**Distance Calculation**' choose '**Smallest Enclosing Rectangle**'.
3. Keep all other settings as default.
4. **Confirm** the process settings.
5. **Execute** the process.

Process Check

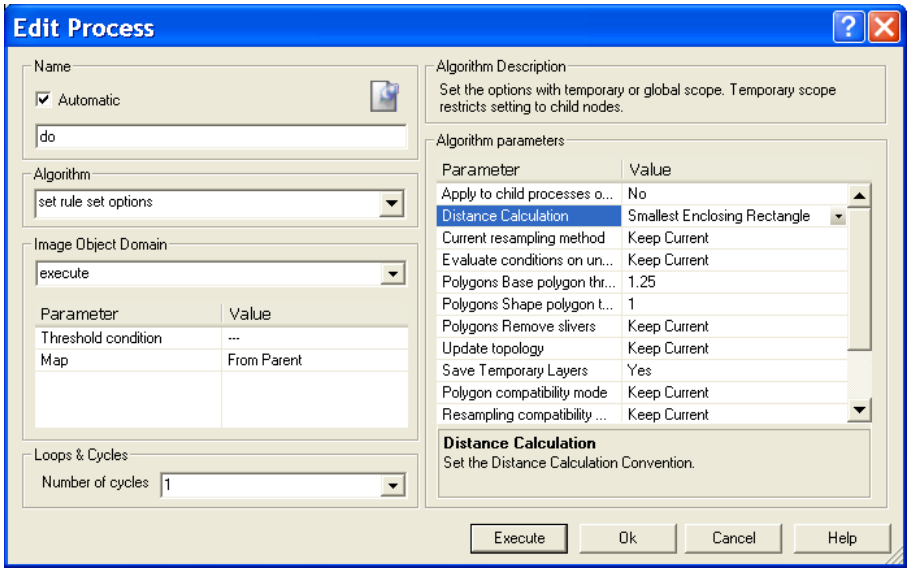


Figure 72: Process settings for defining distance calculation mode.

9.5.5 Create the customized relational feature expressing spectral differences/similarities

Rule Set Check

Introduction to the customized feature

Introduction

The operator

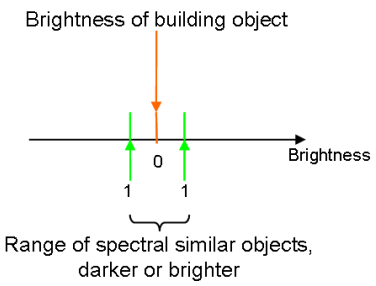
When creating the customized relational feature carefully make up your mind about the operators to use.

There are two operators available expressing differences:

- The operator '**Mean difference**', which can also give back negative values for objects darker than the building object.
- The operator '**Mean absolute difference**', which gives back only positive values.

A rule must be formulated, expressing 'similarity', no matter whether slightly brighter or darker. Therefore the operator '**Mean absolute difference**' is used, as this operator gives back only **positive values**.

Introduction



The search range for neighboring objects

With the customized relational features, there is the possibility to define the search range in which objects are evaluated according to their similarity. This means not all objects are compared to a 'Building' object, but only those object within a certain distance. Here in our case a distance of **20 pixels** is sufficient.

Create the customized relational feature 'MeanAbsDiff'

Action!

1. In the 'Feature View' window browse to 'Object features>Customized' and double-click on '**Create new 'Relational Feature''**.

The 'Edit Customized Feature' dialog box opens.

2. Name the feature '**MeanAbsDiff_CustBrightness_20_Buildings**'.
3. As operator choose '**Mean absolute difference**' (1).
4. As feature choose '**Customized_Brightness**' (2).
5. As distance insert **20** (3)
6. As Class choose **'Building'** (4).
7. Confirm the settings with 'OK'.

The feature is added to the Feature View, you can find it under: 'Class-Related features>Customized'.

Settings Check

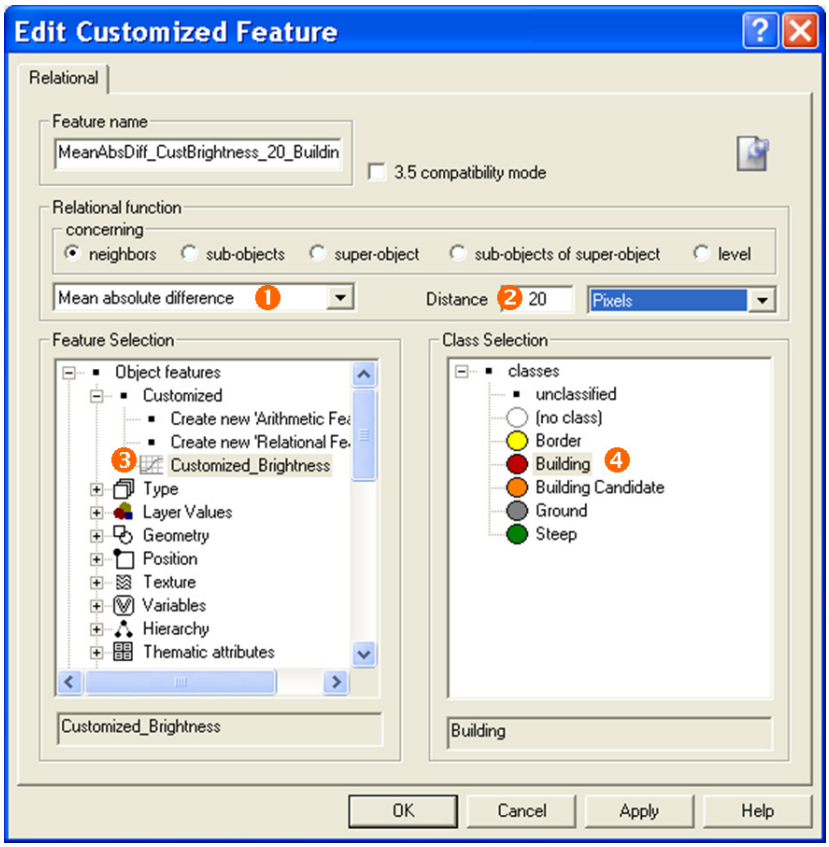


Figure 73: The 'Edit Customized Feature' dialog box with settings to create the feature 'MeanAbsDiff_CustBrightness_20_Buildings'.

9.5.6 Find a threshold to classify spectrally similar 'Building Candidates' to the class 'Buildings'

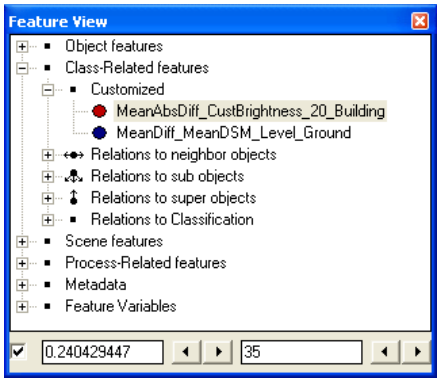
Action!

1. In the 'Feature View' window browse to '**Class-Related features>Customized**'.
2. Double-click on the feature '**MeanAbsDiff_CustBrightness_20_Buildings**'.

> **Note:**
>
> Objects highlighted red means that they are undefined. In our case, they are more than 20 pixel away from a 'Building' object.

High values mean high difference to 'Building' objects, low values mean low difference to 'Building' objects.

3. Right-click on the above mentioned feature and '**Update Range**' from the context menu. Decrease the maximum value to 35.



Check Feature View

Figure 74: The feature '**MeanAbsDiff_CustBrightness_20_Buildings'** is added to the feature list.

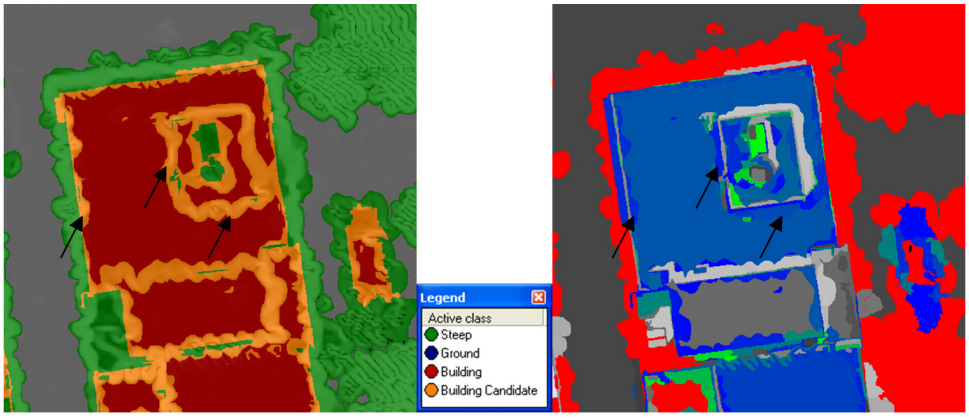


Figure 75: Spectrally too different objects have a higher threshold than 35 for the feature 'MeanAbsDiff_CustBrightness_20_Buildings'.

9.5.7 Classify using the Customized Class-Related feature

Action!

1. Select the last process and right-click on it and select 'Append New'. Choose '**assign class'** as algorithm.
2. In the Class filter choose 'Building Candidates'.
3. Click on the 'no condition' button and browse to the feature '**Class-Related Features>Customized>MeanAbsDiff_CustBrightness_20_Buildings**'.
4. Insert **lower than 35** in the 'Edit threshold condition'.
5. As 'Use class' define '**Building**'.
6. **Execute** the process.

Settings Check

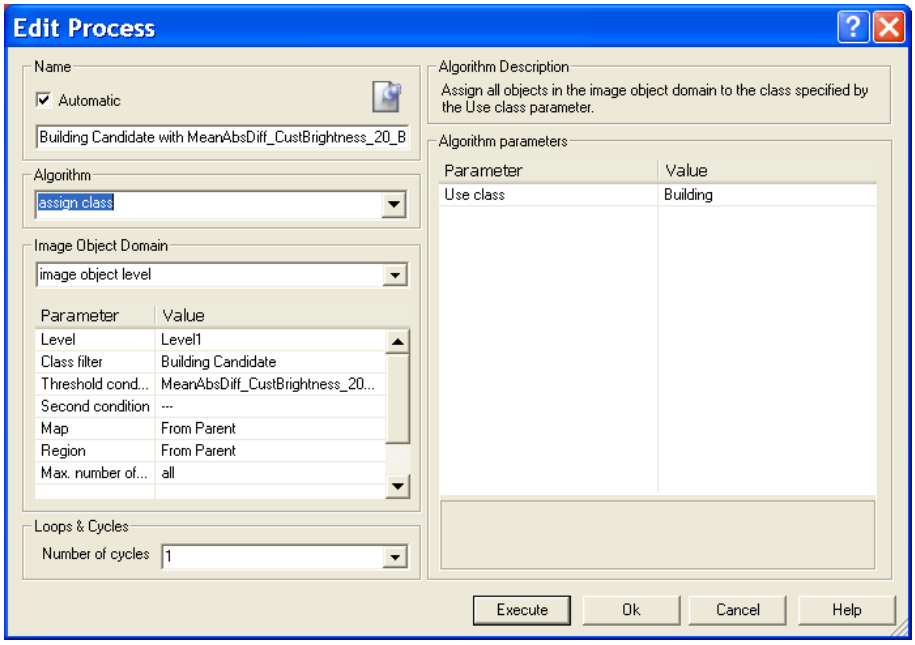


Figure 76: Process settings for classifying spectrally similar objects to the class 'Building'.

Rule Set Check

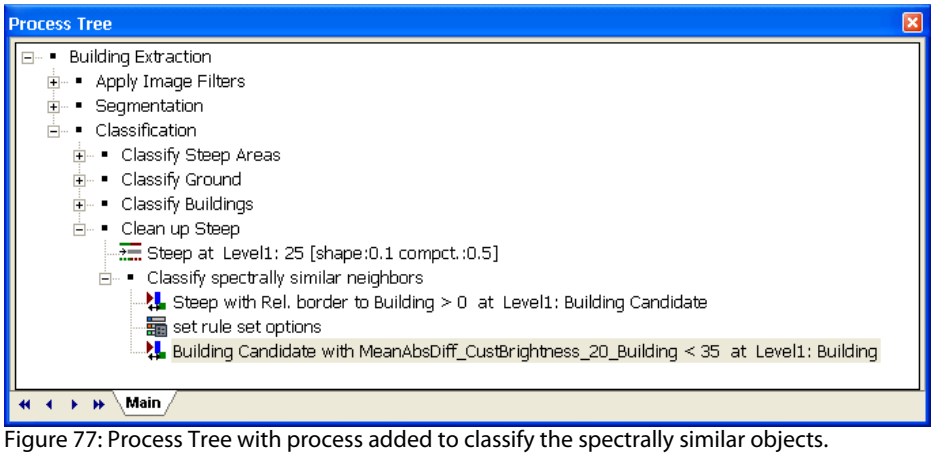


Figure 77: Process Tree with process added to classify the spectrally similar objects.

Review intermediate result

Some 'Building Candidate' objects were not classified as 'Building'. Their spectral difference was too high, e.g. the shadows.

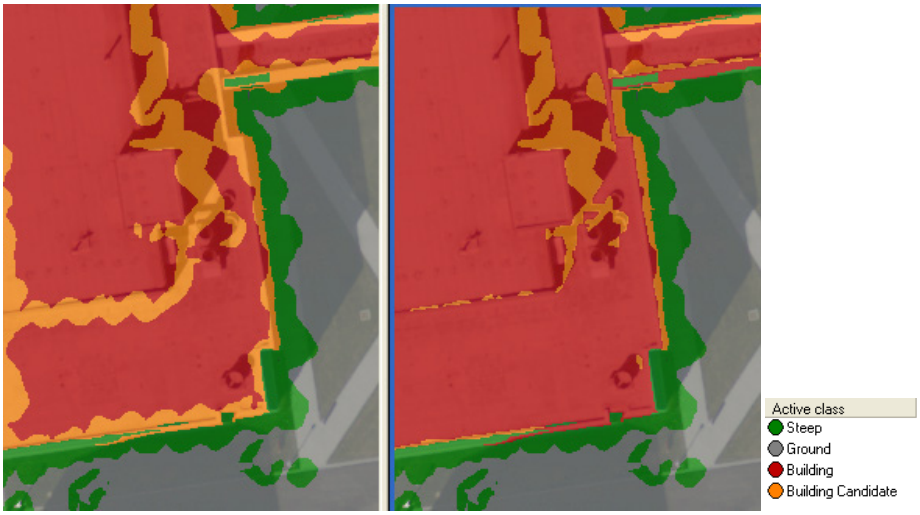


Figure 78: Left: Classification view before spectral similarity classification, Right: after classification.

Result Check

9.5.8 Classify the temporary class 'Building Candidates' back to the class 'Steep'

1. Select the last process, right-click on it and select 'Append New'. Choose **'assign class'** as algorithm.
2. The process has to be applied for 'Building Candidates' (choose it in the Class filter field).
3. Keep 'no condition'.
4. For the 'Use class' parameter set click in the 'Value' field and select 'Steep'.
5. Execute the process.

Action!

All remaining 'Building Candidate' objects are classified back to the class 'Steep' again.

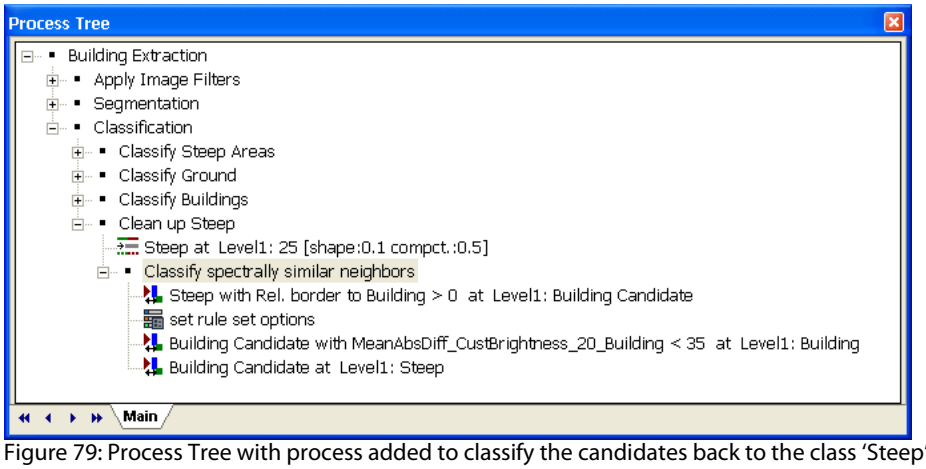
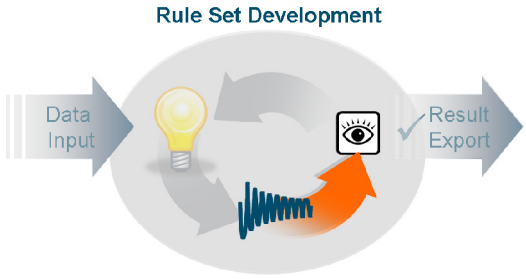


Figure 79: Process Tree with process added to classify the candidates back to the class 'Steep'.

Rule Set Check

Next step: Review Result!



9.6 Review the result – spectrally similar objects are classified

'Steep' objects which are spectrally similar to adjacent 'Building' objects were classified as 'Building', too.

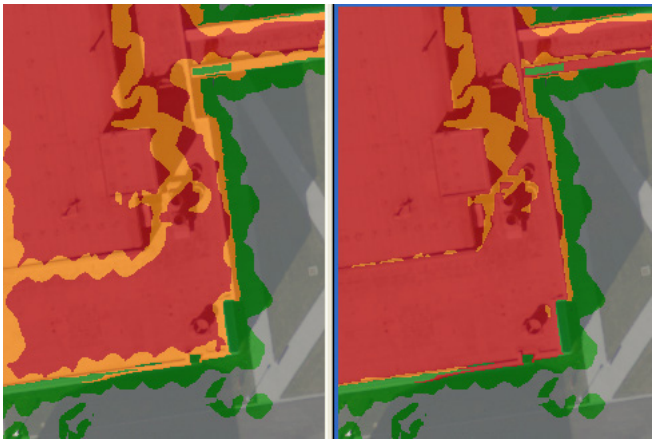


Figure 80: Before and after the spectral similarity classification.

Result Check

Still not all 'Steep' objects which actually belong to the class 'Building' are classified. Some of them are **highly surrounded by the class 'Building'**. These objects has to be classified as 'Buildings', too to fill the holes.

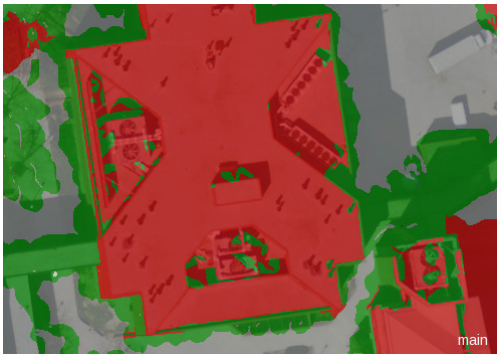
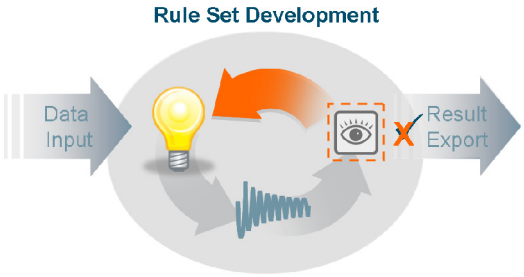


Figure 81: Classification result after 'Building Candidates' are reclassified as 'Steep'.

Next step: Develop strategy for next analysis step!



9.7 Strategy to fill holes in 'Building' classification

This chapter covers the following content

- ➔ *'Looping' a process – automatically repeat the procedure*
- ➔ *Find totally enclosed objects*

Introduction

Within the 'Building' objects there are still some 'Steep' objects classified.

The existing 'Building' objects will be grown into 'Steep' objects, but only in those which are highly surrounded by the class.

9.7.1 'Looping' a process – automatically repeat the procedure

In a first step those objects are classified, which have a high relative border to 'Building' objects.

Figure 82: First neighbors with a high common border are classified.

But if this is executed only once, the gaps in the 'building' objects are not yet closed. After one round, there are new 'Building' objects with again some 'Steep' objects having a high relative border to them. To grow the 'Building' objects into all objects having a high relative border the process has to be **re-executed** until all are swallowed. This can be achieved by setting a '**Loop**'.

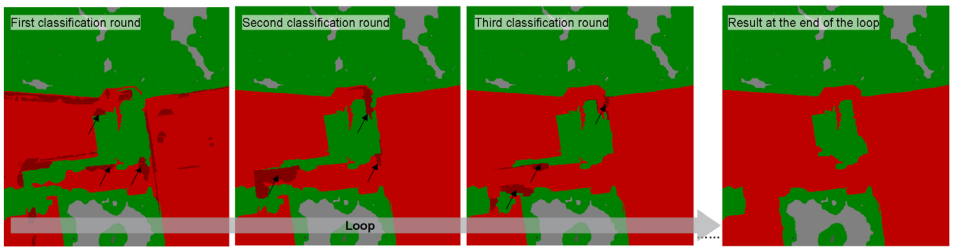


Figure 83: Diagram of a looped process to grow the 'Building' objects.

9.7.2 Find totally enclosed objects

After those with a high relative border are classified 1 a second step is necessary to identify those objects totally surrounded 2 by 'Building' objects.

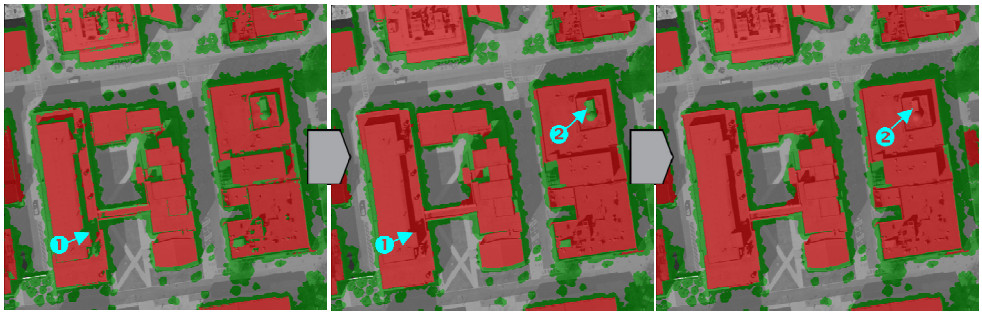
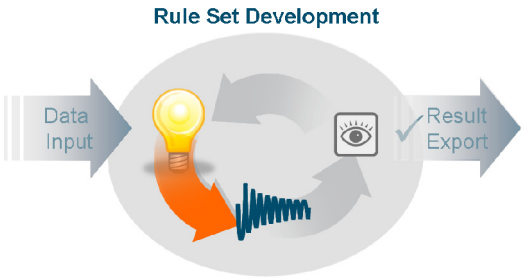


Figure 84: Left: Before first processing step, Middle: after first processing step; Right: After second processing step.

Next step: Translate Strategy into Rule Set!



9.8 Translating strategy – find surrounded and enclosed

This chapter covers the following content

- ➔ *Classify those with a high relative border to 'Building' objects*
- ➔ *Review intermediate result*
- ➔ *Classify those totally surrounded using algorithm 'find enclosed by class'*

Information

9.8.1 Classify those with a high relative border to 'Building' objects and loop the process

As already explained in chapter 9.5.1 the feature 'Rel. border to' will be used to express the adjacency to 'Building' objects.

Find the threshold for classification

The objects which will be classified during the looped sequence must be restricted. The objects shall have at least a **50% border to 'Building' objects**, this helps to avoid that the growing 'building' classification is growing too far.

Action!

1. In the 'Feature View' window browse to 'Class-Related Features>Relations to neighbor objects'.
2. Double-click on the feature '**Rel. border to 'Building**''.
3. Right-click and '**Update Range**' from the context menu. Increase the minimum value to **0.5**.

All objects with more than 50% common border to 'Building' objects are highlighted in a range from blue to green.

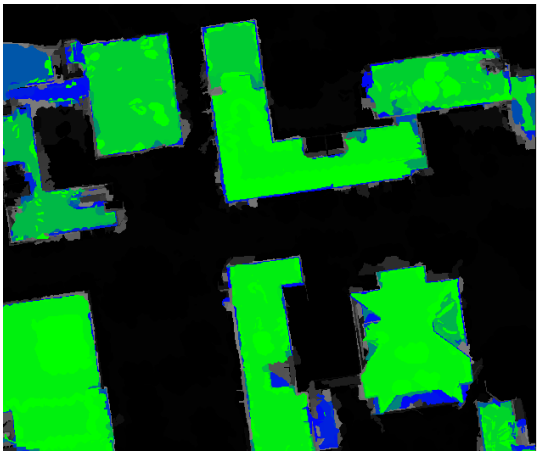


Figure 85: Feature View with feature 'Rel. border to 'Building'' updated to value larger than 0.5.

Check Feature View

Classify using the feature 'Rel. border to'

Prepare process structure

1. Collapse the '**Classify spectrally similar neighbors**' process and right-click on it. Select 'Append New' and name the process '**Find highly surrounded**'.

Action!

Insert classification process

2. Select the parent process, right-click on it and select 'Insert Child'. Choose '**assign class'** as algorithm.
3. Set the Class filter to 'Steep'.
4. Insert the condition '**Rel. border to 'Buildings''** larger than 0.5.
5. In the section 'Loops & Cycles' select '**Loop while something changes**' from the drop-down list.
6. Define '**Building**' as 'Use class'.
7. **Execute** the process.

Settings Check

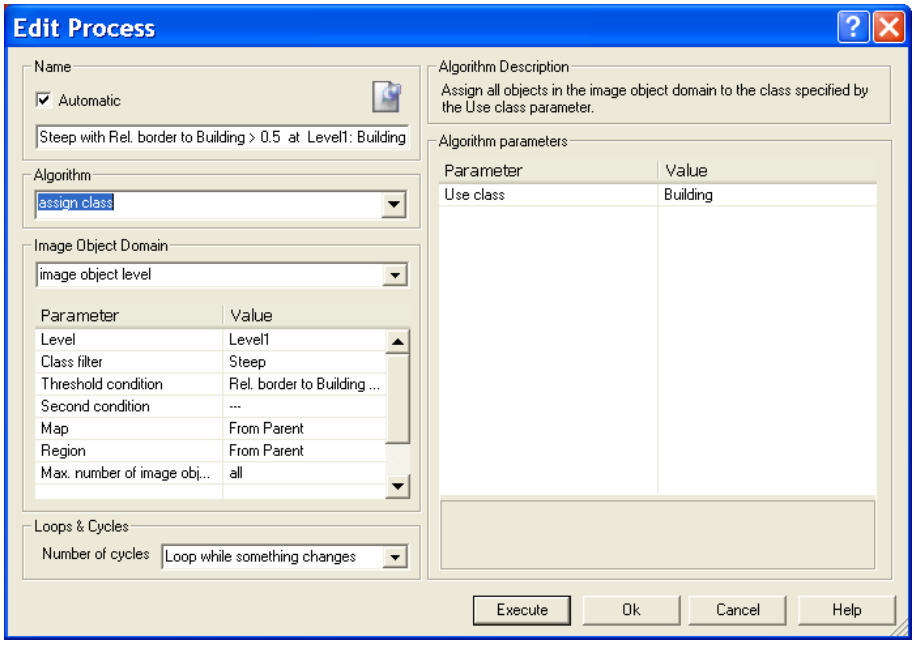


Figure 86: Process settings to classify 'Steep' objects with a high relative border to 'Building' objects as 'Building'.

Rule Set Check

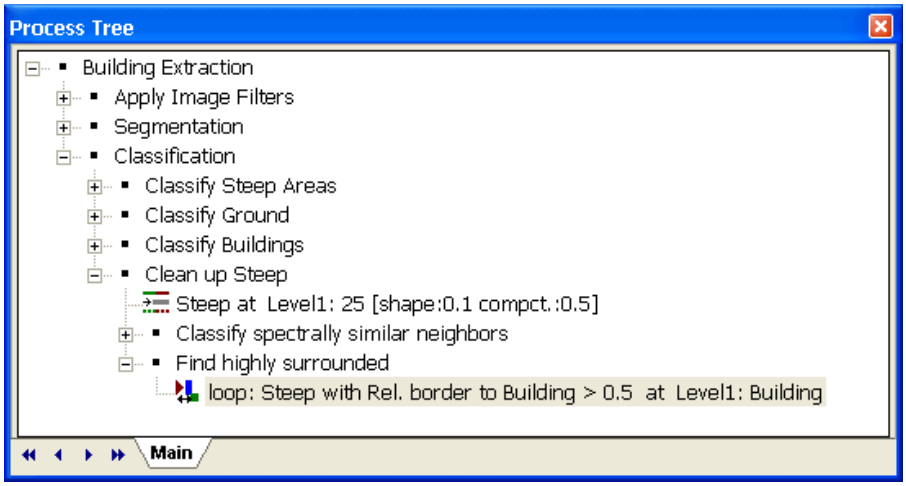
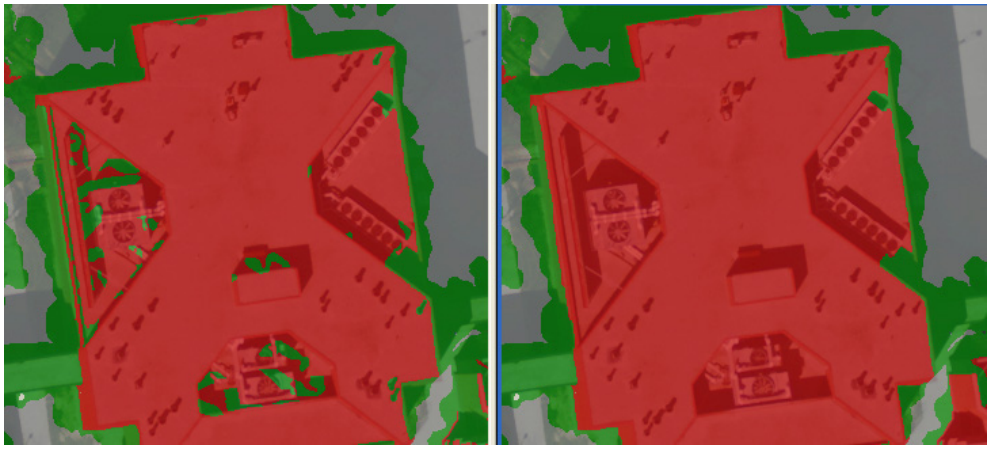


Figure 87: Process Tree with process added to classify highly surrounded 'Steep' objects.

9.8.2 Review intermediate result

The 'Steep' objects with high relative border have been iteratively classified as 'Buildings'. Still some totally surrounded, which had no high relative border to 'Building' objects still remain. They consist of several objects.



Result Check

9.8.3 Classify those totally surrounded using algorithm 'find enclosed by class'

The algorithm 'find enclosed by class'

Introduction

eCognition offers a **special algorithm to identify objects enclosed by other classes**. In contrast to the feature 'Relative Border to' this algorithm also classifies objects as 'enclosed', if they are not direct neighbors of the enclosing class, i.e. if they consist of several objects.

Introduction

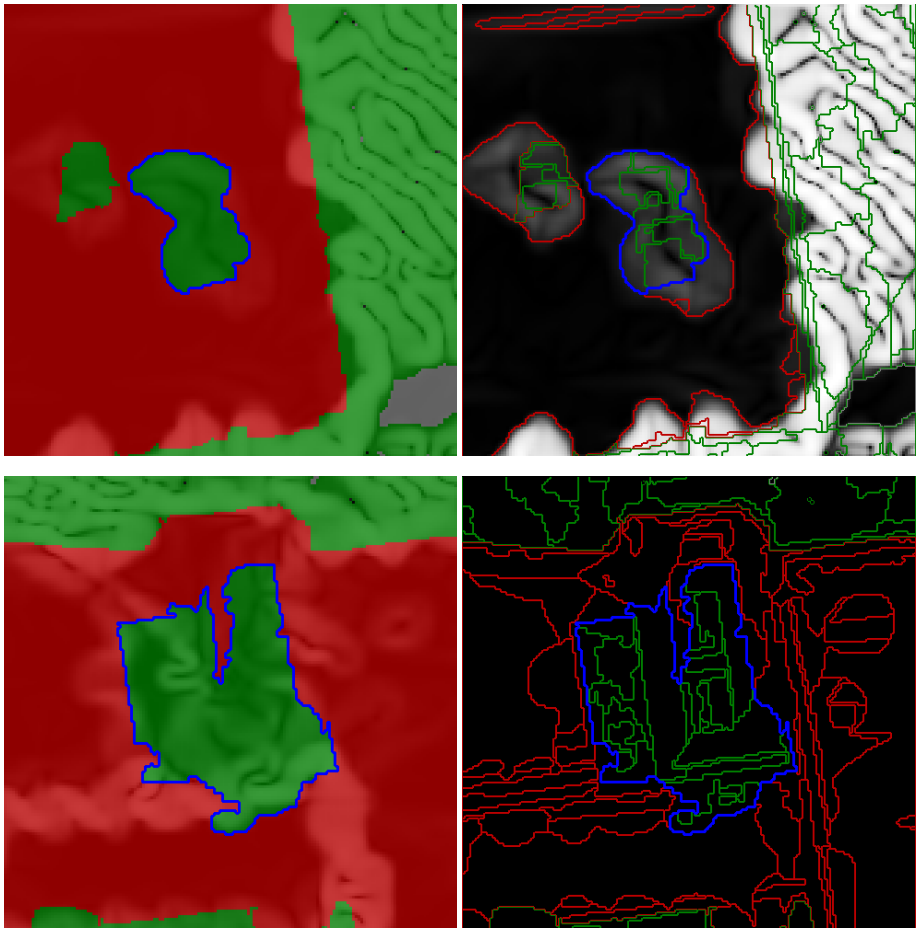


Figure 88: Selected objects are enclosed by 'Building' objects.

Note:

Objects at the image border are not classified as **enclosed** class. The shared part of the outline with the image border will not be recognized as enclosing border. To get such kind of object use the algorithm 'find enclosed by image object'.

Insert process to classify all enclosed 'Steep' objects as 'Building'

Action!

1. Select the last process, and right-click on it and select 'Append New'. Choose '**find enclosed by class'** as algorithm.
2. Change the 'Class filter' to '**Steep**'.
3. In the field '**Enclosing classes**' define the class that encloses, here '**Building**'.
4. In the field 'Active classes' define the class to which the enclosed object shall be classified, here again '**Building**'.
5. Keep all other default settings.
6. **Execute** the process.

All 'Steep' objects which are enclosed by 'Building' objects are now classified as 'Building', too.

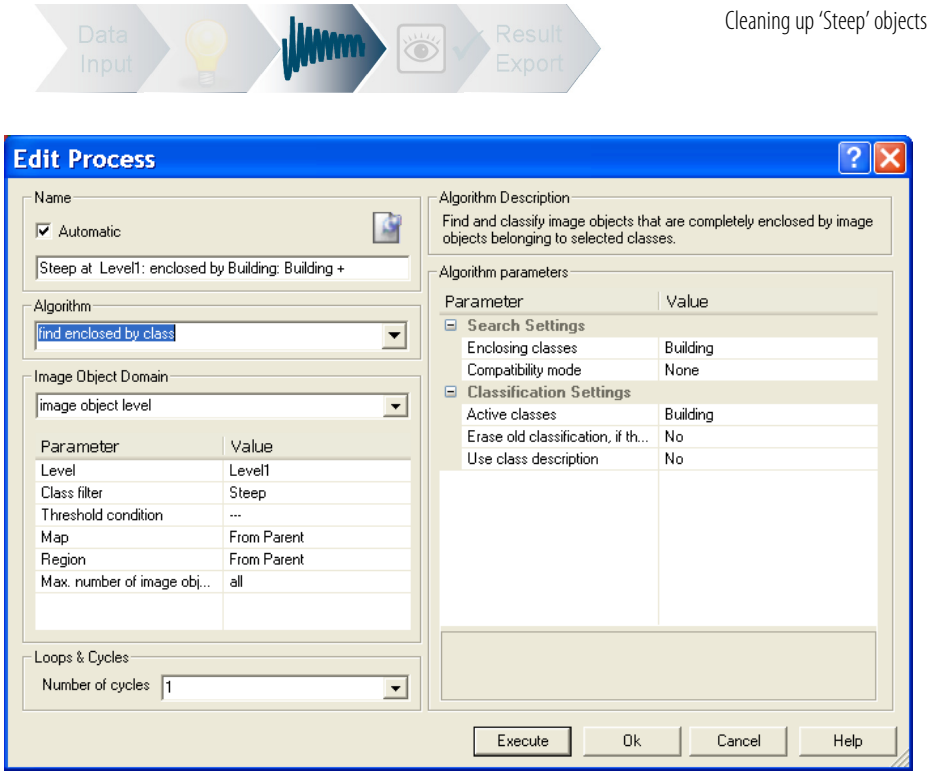


Figure 89: Process Settings to classify 'Steep' objects enclosed by 'Building' objects.

Settings Check

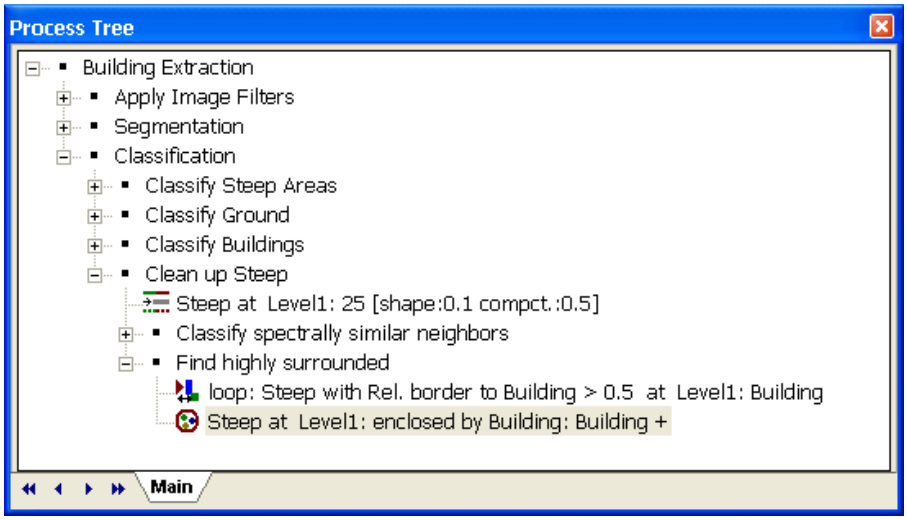
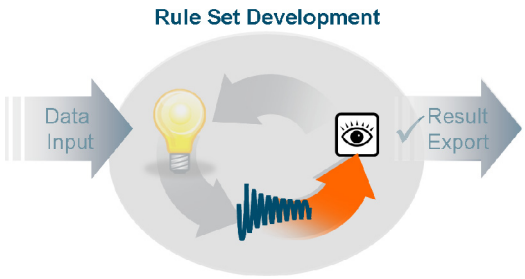


Figure 90: Process Tree with process added to classify 'Steep' objects surrounded by 'Building' objects.

Rule Set Check

Next step: Review Result!



9.9 Review result – enclosed objects are classified

Those 'Steep' objects with a very high relative border (more than 50%) and those completely surrounded are now classified as 'Building', too.

Result Check

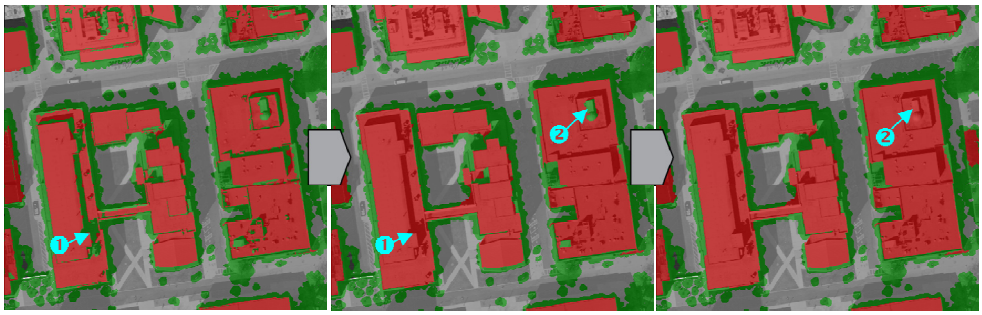


Figure 91: Left: Before first processing step, Middle: after first processing step; Right: After second processing step.

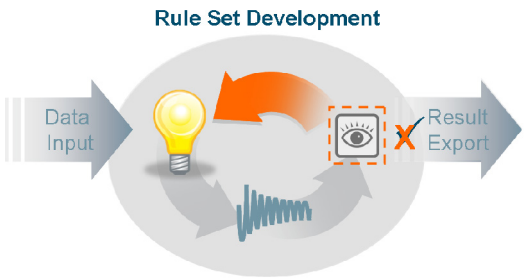
Some vegetated areas are still misclassified as 'Building'.

Figure 92: Classification view of the whole scene. The arrows point to vegetated areas which are misclassified as 'Building'.

Lesson 9 covered the following content

- *Strategy to reshape objects*
- *Translating strategy - reshape using the spectral layers*
- *Review result – objects are reshaped*
- *Strategy to grow into spectrally similar objects*
- *Translating strategy – classify adjacent, spectrally similar objects*
- *Review the result – spectrally similar objects are classified*
- *Strategy to fill holes in 'Building' classification*
- *Translating strategy – find surrounded and enclosed*
- *Review result – enclosed objects are classified*

Next step: Develop strategy for next analysis step!



Lesson 10 Cleaning up 'Building' objects

This lesson covers the following content

- *Strategy to clean up vegetation misclassified as 'Building'*
- *Translating 1st step: use spectral feature to declassify vegetated areas*
- *Review result of first step – the big vegetation objects are gone*
- *Strategy to get rid of again enclosed*
- *Translating 2nd step – copy and paste existing 'find enclosed by class' process*
- *Review result*
- *Strategy to declassify too small objects*
- *Translating 3rd step: declassify the too small*
- *Review result*

Introduction

10.1 Strategy to clean up vegetation misclassified as 'Building'

This lesson to clean up the 'Building' objects has three major parts.

- First: the misclassified vegetation is eliminated

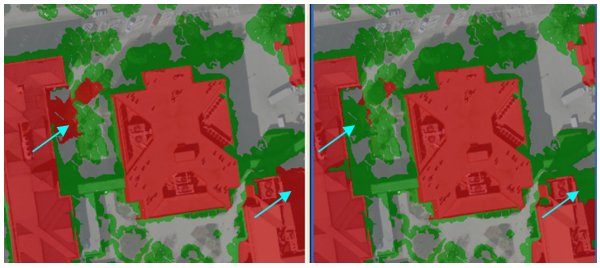


Figure 93: Classification view before and after the refinement. Arrows pointing on misclassified areas which are now correctly classified.

- Second: the again enclosed 'Steep' objects are declassified

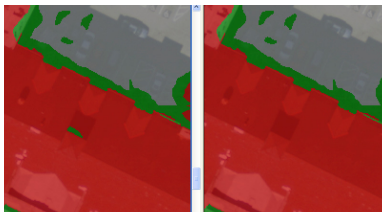


Figure 94: The enclosed objects are reassigned to the class 'Building'.

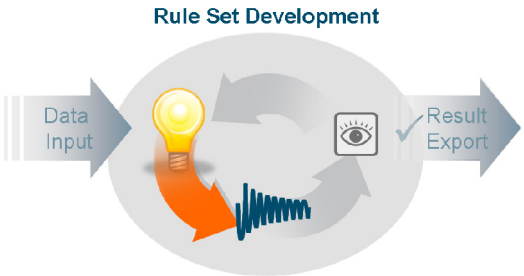
- Third: the too small 'Building' objects are declassified

Introduction

As discussed in the Module 1 of the LiDAR Guided Tour (Lesson 1.8) a characteristic feature for classifying vegetation with RGB data is to compare the **green layer** to all spectral layers, respectively to use a customized feature 'Ratio Green'.

Still a second step is needed to declassify small objects which will remain. As already done in Lesson 8 Refining the 'Building' classification again the 'Area' feature will be used. As the actual 'Building' objects now consist of several objects, first they must be merged.

Next step: Translate Strategy into Rule Set!



10.2 Translating 1st step: use spectral feature to declassify vegetated areas

This chapter covers the following content

➔ *Load the customized feature 'CustRatioGreen'*

➔ *Create a process to declassify vegetation misclassified as 'Building' objects*

10.2.1 Load the customized feature 'CustRatioGreen'

Action!

1. Right-click in the Feature View and select '**Load**'.
2. Browse to the folder 'Rule Sets' in your Guided Tour directory and select '**CustRatioGreen.duf**'.

Result Check

The customized feature is added to the Feature View under the category 'Object features>Customized'.

As in the Guided Tour 1, the vegetation objects have a value higher than 0.36 for the feature 'Cust_Ratio_Green'.

10.2.2 Create a process to declassify vegetation misclassified as 'Building' objects

Prepare the Rule Set structure

Action!

1. In the process Tree **right-click** on the 'Clean up 'Steep'' parent process and select '**Append New**' from the context menu.
2. Insert '**Clean up 'Building''** in the 'Name' field and confirm with OK.

Insert process to classify

1. **Right-click** on the inserted parent process and select '**Insert Child**' from the context menu.
2. Choose '**assign class**' from the algorithm list.
3. Select '**Buildings**' as 'Class filter'.
4. For 'Threshold condition' browse to 'Object features>Customized>**CustRatioGreen**', double-click on it.

5. In the 'Edit threshold condition' choose **larger than (>) 0.36**.
6. In the field 'Use class' select '**Steep**'.
7. Confirm with **OK**.
8. **Execute** the process.

All 'Building' objects with 'Customized Ratio Green' higher than 0.36 are now classified as 'Steep' again.

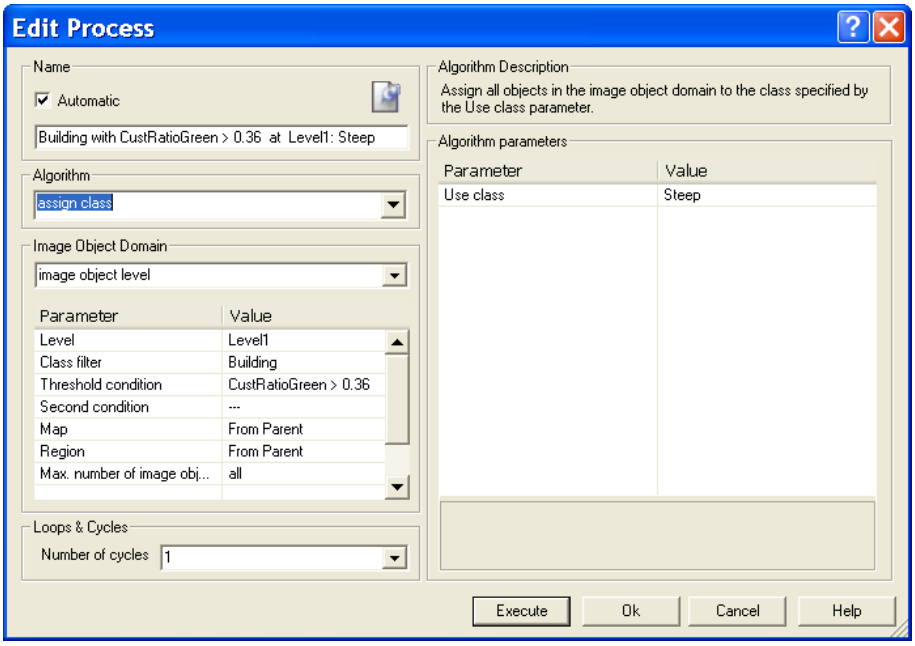


Figure 95: 'Edit Process' dialog box with settings to classify 'Building' objects with a value for the customized feature higher than 0.36 back to the class 'Steep'.

Settings Check

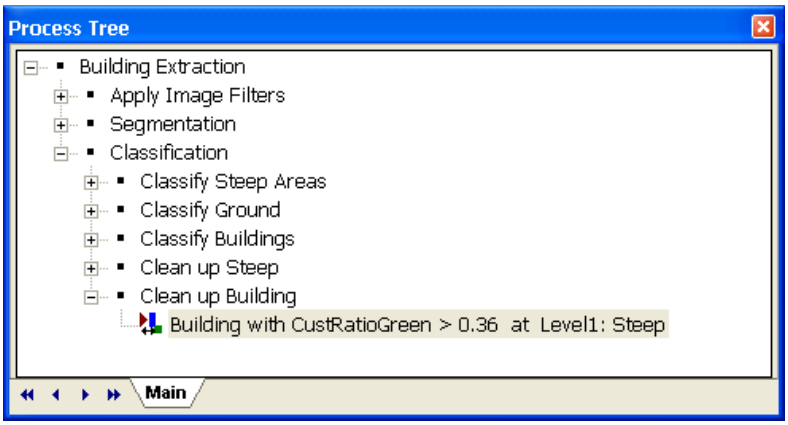
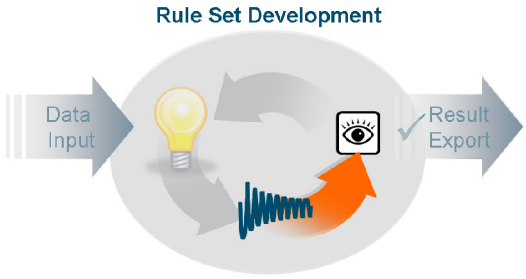


Figure 96: Process Tree with processes added.

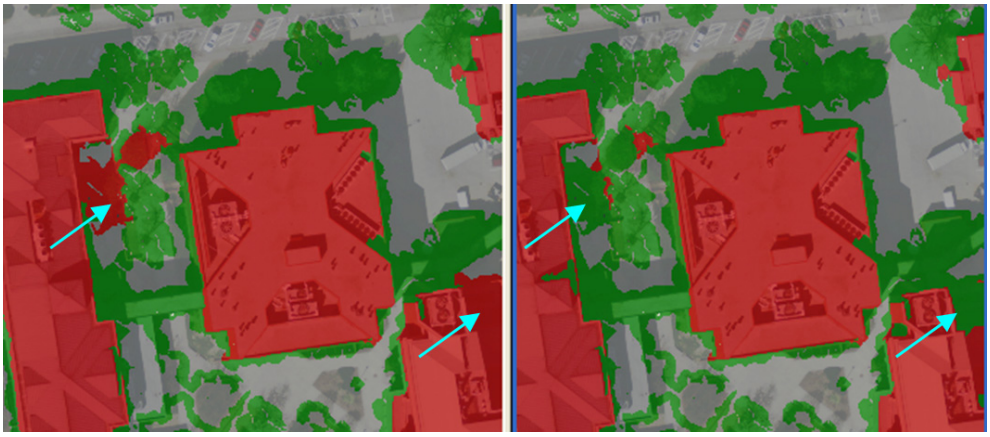
Rule Set Check

Next step: Review Result!



10.3 Review result of first step – the big vegetation objects are gone

Misclassified vegetated areas are now classified as 'Steep'.



Result Check

Figure 97: Classification view before and after the refinement. Arrows pointing on misclassified areas which are now correctly classified.

10.4 Strategy to get rid of again enclosed

Introduction

After classifying with the customized ratio of green there are also some objects inside the buildings, again classified as 'Steep'. To get rid of those, again the 'find enclosed by class' algorithm is used. Simply copy the existing process, paste and execute it.

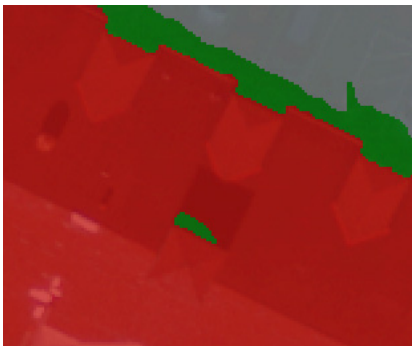


Figure 98: Again 'Steep' objects appear within 'Building' objects.

10.5 Translating 2nd step – copy and paste existing 'find enclosed by class' process

Action!

1. Go to the process '**Steep at Level1: enclosed by Building: Building +**' in the Rule Set Section 'Find highly surrounded'.
2. Copy and paste the process after the last process.
3. Execute it.

Rule Set Check

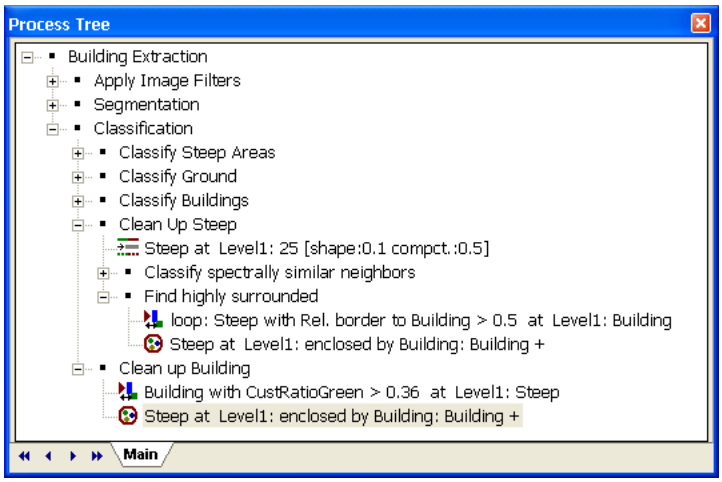


Figure 99: Process Tree with copied and pasted process.

10.6 Review result

Result Check

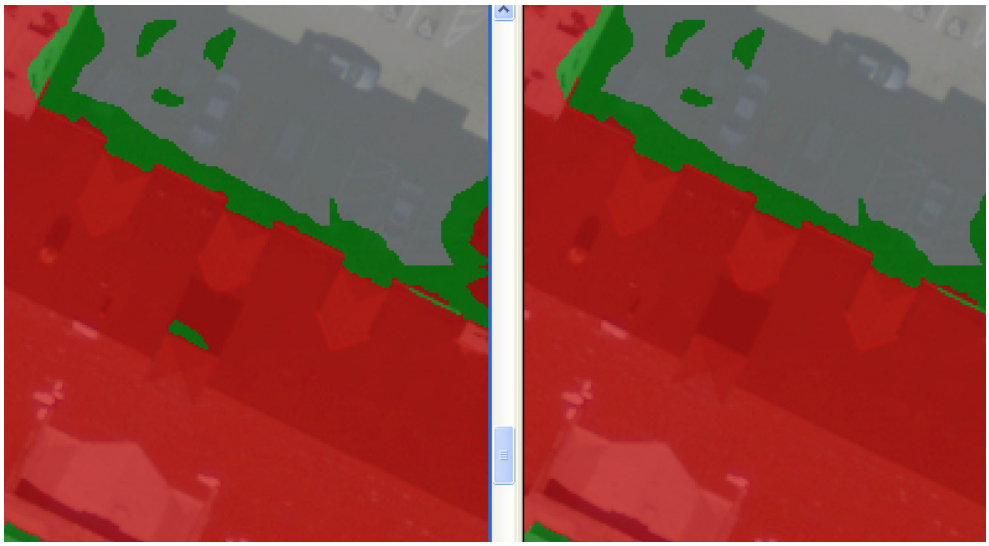


Figure 100: The enclosed objects are reassigned to the class 'Building'.

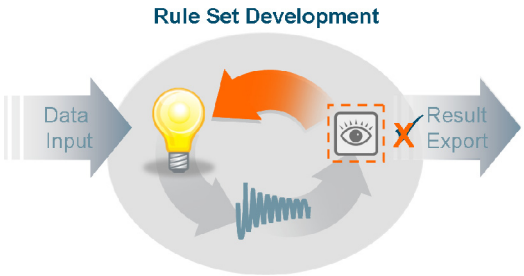
10.7 Strategy to declassify too small objects

Introduction

Here in this case there are still some small objects misclassified. Most of them are actually part of trees. Therefore they should be classified to the class 'Steep', too. The '**Image Object Table'** tool is used to define the size threshold for 'Buildings'. This threshold is increased a bit to have a buffer for classification. Before evaluating with this tool, the 'Building' objects are **merged**.

Figure 101: Small objects within trees, which are misclassified as 'Building'.

Next step: Develop strategy for next analysis step!



10.8 Translating 3rd step: declassify the too small

This chapter covers the following content

- ➔ *Create process to merge all 'Building' objects*
- ➔ *Use the 'Image Object Table' tool to review the smallest objects*
- ➔ *Sort the values and browse to the individual objects*
- ➔ *Insert process to refine classification by using the 'Area' feature*

10.8.1 Create process to merge all 'Building' objects

Action!

1. Right-click on the last process and select 'Append New' from the context menu.
2. Select 'merge region' form the algorithms' list.
3. Make sure that 'Level1' is selected in the Image Object Domain.
4. In Class filter choose 'Buildings'.
5. Execute the process.

All 'Building' objects are now merged.

Settings Check

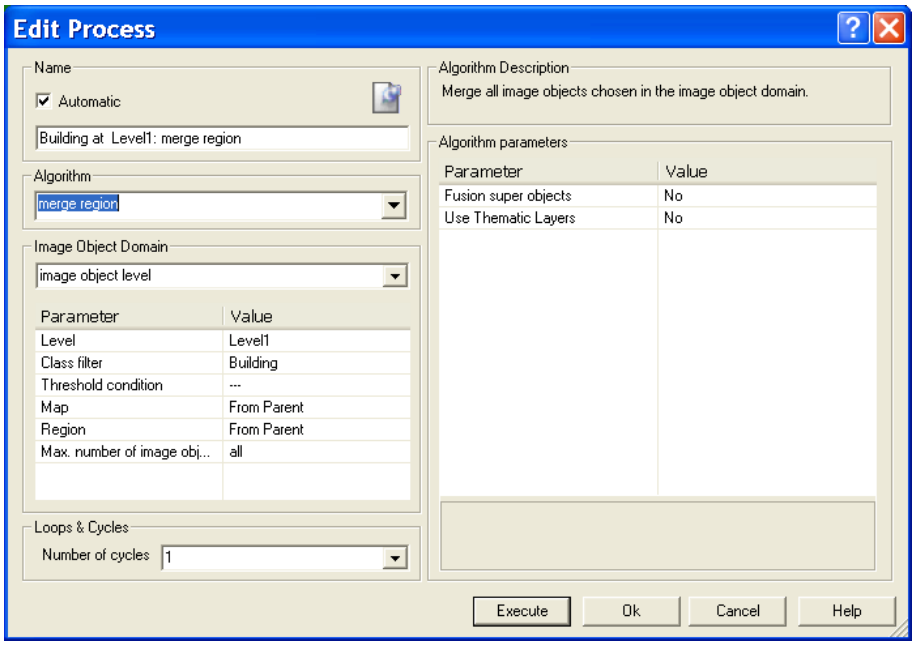


Figure 102: Process settings for merging all 'Building' objects.

10.8.2 Use the 'Image Object Table' tool to review the smallest objects

Information

The '**Image Object Table**' window allows you to sort the image objects according to feature values and by clicking on an object listed in the table, it will be automatically displayed in the viewer.

The 'Image Object Table' is a very useful tool to examine object by object without panning manually.

You may want to compare image objects of selected classes in cases of:

- Reviewing results: evaluation of classifications, debugging and quality assurance.
- As support for finding thresholds for classification, selection of image objects according to sorted feature values.

Configure the table to list all 'Building' objects and their according size

Action!

1. To open the '**Image Object Table**' window, do one of the following:
   - Choose **'Image Objects > Image Object Table'** on the main menu bar

   or

   - Choose '**View > Windows > Image Object Table**' on the main menu bar
2. Right-click in the text area of the 'Image Object Table' window and choose '**Configure Image Object Table…'** from the context menu.

The '**Configure Image Object Table'** dialog box opens.

3. Hit the '**Select Classes**' button to display all its image objects.

The 'Select Classes for List' dialog box opens.

4. Select the class 'Building' by **double-clicking** it and confirm with 'OK'.
5. Click on the '**Select Features'** button.

The 'Select Features' dialog box opens.

6. Browse to the feature '**Object features>Geometry>Extent>Area**', double-click on it and confirm with 'OK'.
7. Click **OK** to display all image objects and their feature values as configured in the '**Image Object Table**'. To each image object an **ID** number is assigned.

Settings Check

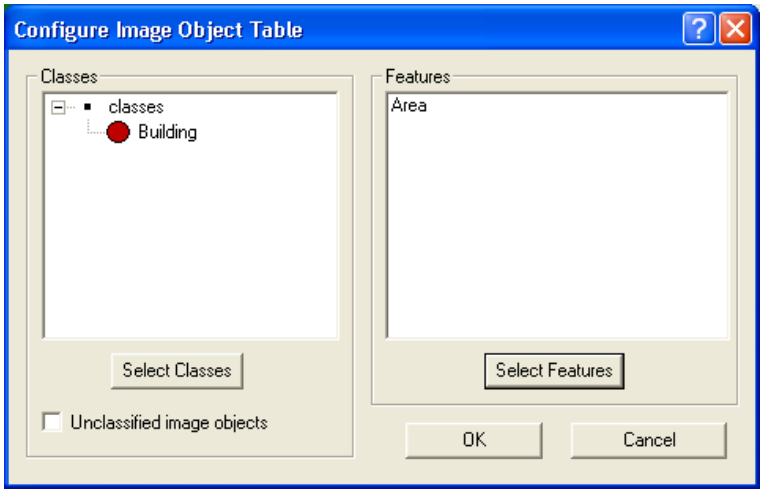
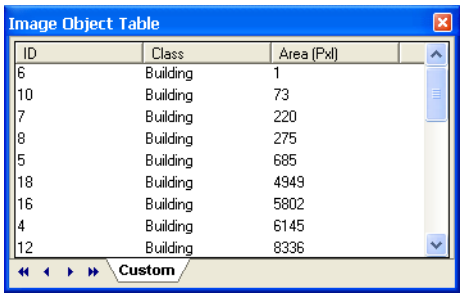


Figure 103: 'Configure Image Object Table' window, with class 'Building' selected and feature 'Area' selected.

Result Check



ID	Class	Area (Pxl)
6	Building	1
10	Building	73
7	Building	220
8	Building	275
5	Building	685
18	Building	4949
16	Building	5802
4	Building	6145
12	Building	8336

Figure 104: 'Image Object Table' window, with class 'Building' selected and feature 'Area'.

10.8.3 Sort the values and browse to the individual objects

The object with size 685 is already a building. All smaller objects must be declassified.

Action!

1. Click the 'Area' column header to sort according to the column values. Make the **smallest value onto** the top of the list.
2. Zoom into the image so that you can see the individual 'Building'.
3. Click on one row in the table to select the corresponding image object.
4. Evaluate the objects from smaller to larger, objects smaller than about 700 are actually no buildings.

Note:

If you want to **change the outline color** of selected objects to see them better go to main menu 'View>Display Mode>**Edit highlight colors**' and change the 'Selection color'. Click on the '**Active view**' button to apply the change.

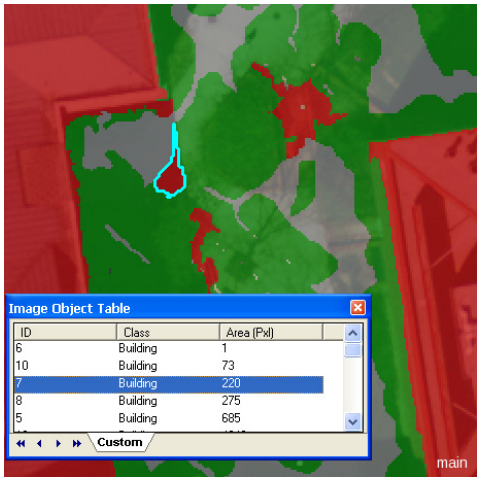


Figure 105: A selected image object is highlighted twice: in the 'Image Object Table' window and in the project view.

Result Check

10.8.4 Insert process to refine classification by using the 'Area' feature

The value for classification is increased from 700 to 1000 to have a buffer in size. If you want to be more strict, you can also use a lower value like 700.

1. Append a new process and choose algorithm 'assign class'.
2. As 'Class filter' define 'Building'.
3. As condition set the feature 'Area' **lower than 1000**.
4. Set 'Use class' to '**Steep**'. Confirm with OK.
5. **Execute** the process.

Action!

Settings Check

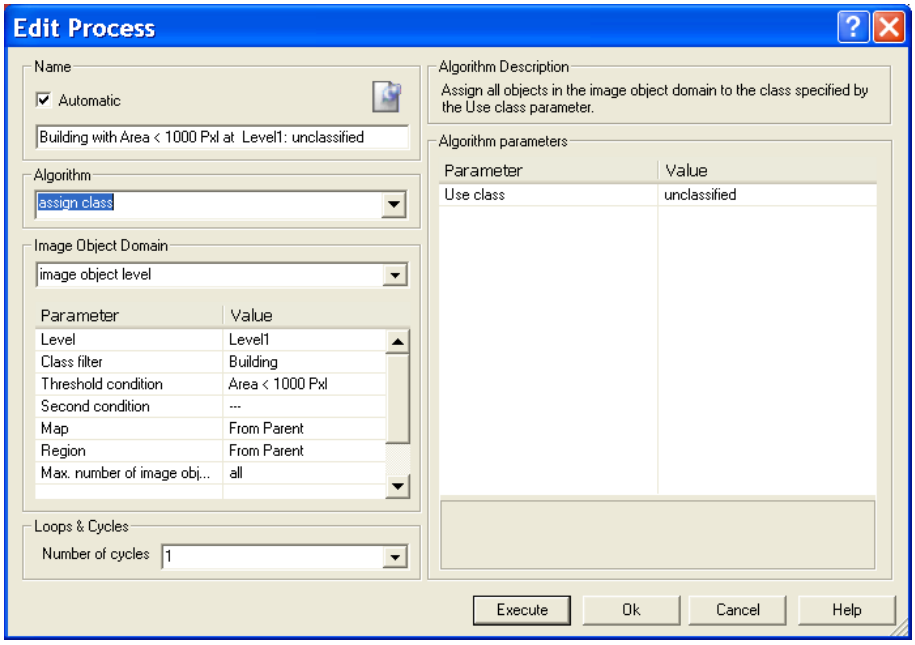
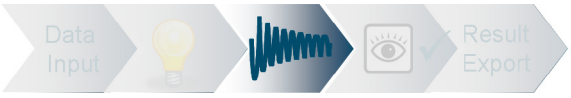


Figure 106: Process settings for refine classification of 'Building' objects.



Rule Set Check

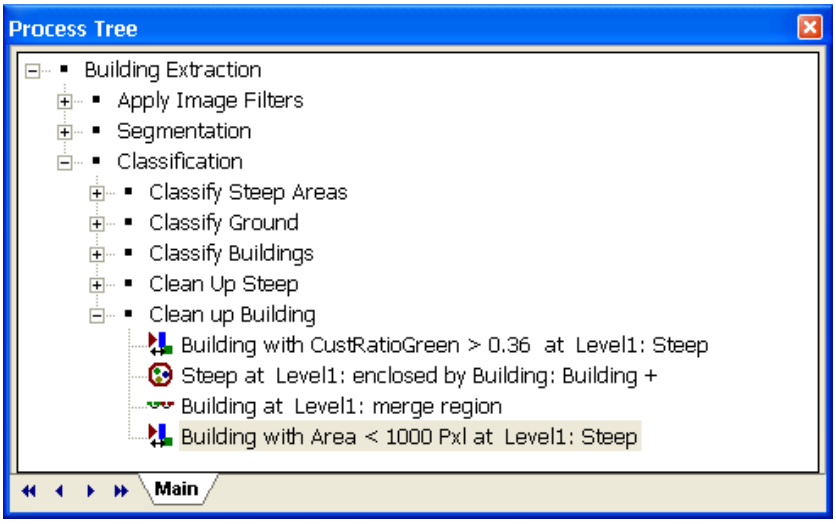
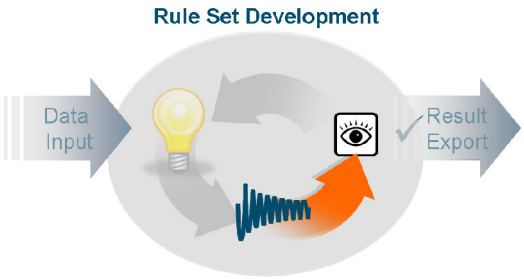


Figure 107: Process Tree with classification process added.

Next step: Review Result!



10.9 Review result

Again check the objects using the 'Image Object Table'as explained in Chapter 10.9.

If you review the classification, the too small objects are now declassified. Most of the buildings have been classified properly.

Result Check

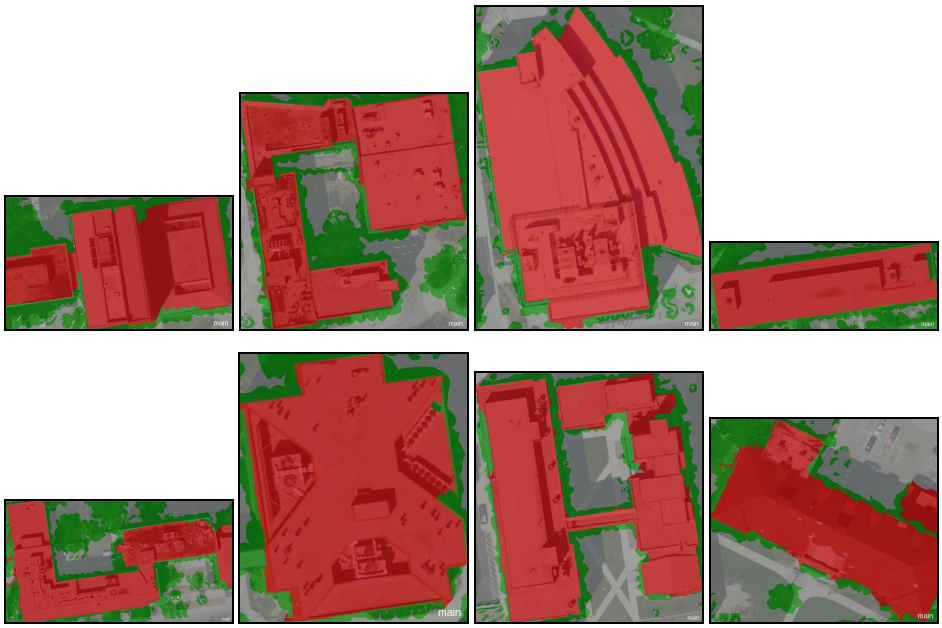


Figure 108: Thumbnails of selected 'Building' objects.

Nevertheless, there are still some parts of the roofs not classified. There are strategies to refine the shape of the buildings, but for the Guided Tour we will accept this level of accuracy.

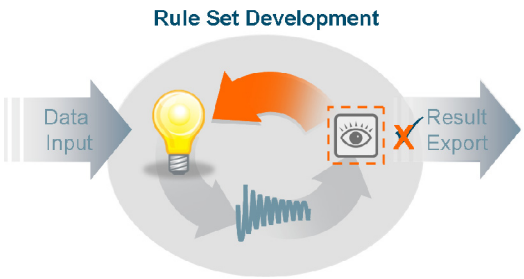
This means that we are now at the point to prepare the export and insert a process to execute the export.

Figure 109: Classification result detail.

Lesson 10 covered the following content

- *Strategy to clean up vegetation misclassified as 'Building'*
- *Translating 1st step: use spectral feature to declassify vegetated areas*
- *Review result of first step – the big vegetation objects are gone*
- *Strategy to get rid of again enclosed*
- *Translating 2nd step – copy and paste existing 'find enclosed by class' process*
- *Review result*
- *Strategy to declassify too small objects*
- *Translating 3rd step: declassify the too small*
- *Review result*

Next step: Develop strategy for next analysis step!



Lesson 11 Preparing the export

This lesson covers the following content

- *Strategy to export the results – declassify those classes not of interest, merge objects*
- *Translating strategy – declassify 'Steep' and 'Ground'*
- *Review result*
- *Translating strategy – remove small objects*
- *Review result*

11.1 Strategy to export the results – declassify those classes not of interest, merge objects

This chapter covers the following content

- *Clean up classification*
- *Get rid of small objects*

Introduction

In this Guided Tour only the information about the buildings will be exported, not for the other classes. Still some very small objects exist. Therefore two major steps have to be implemented:

- First: **clean up the classification**
- Secondly: **get rid of the small** objects

11.1.1 Clean up classification

After the small objects are removed, the classification will be reduced only to 'Building' and 'unclassified' objects.

- **declassify the 'Steep' and 'Ground'** objects
- **merge all declassified** objects to avoid exporting objects not of interest

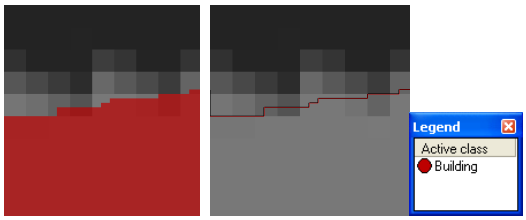


Figure 110: Left: Classification view; Right: Outlines view. Both after the execution.

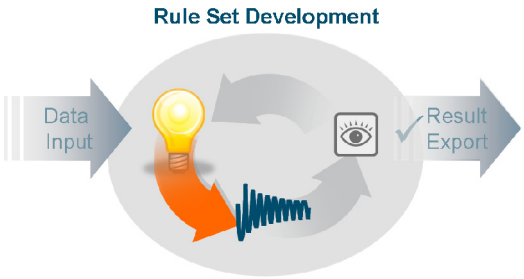
Introduction

11.1.2 Get rid of small objects

To avoid exporting unnecessary information and vector outlines of objects that are not of interest, a sequence of processes is added.

- The first step is to **merge** all 'unclassified' objects.
- Then a special algorithm '**remove objects**' is used to incorporate the smaller 'unclassified' objects to the neighboring object with the largest common border.

Next step: Translate Strategy into Rule Set!



11.2 Translating strategy – declassify 'Steep' and 'Ground'

This chapter covers the following content

- ➔ *Prepare the Rule Set structure*
- ➔ *Insert process to declassify*
- ➔ *Create process to merge all 'unclassified' objects*

11.2.1 Prepare the Rule Set structure

Action!

1. In the Process Tree **right-click** on the 'Clean up 'Building'' parent process and select '**Append New**' from the context menu.
2. Insert '**Prepare Export**' in the 'Name' field and confirm with OK.

11.2.2 Insert process to declassify

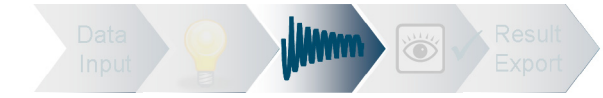
1. Insert a child process and choose algorithm '**assign class**'.
2. As 'Class filter' define '**Steep**' and '**Ground**'.
3. Keep 'no condition'.
4. In the 'Use class' field keep '**unclassified**'. Confirm with '**OK**'.
5. **Execute** the process.

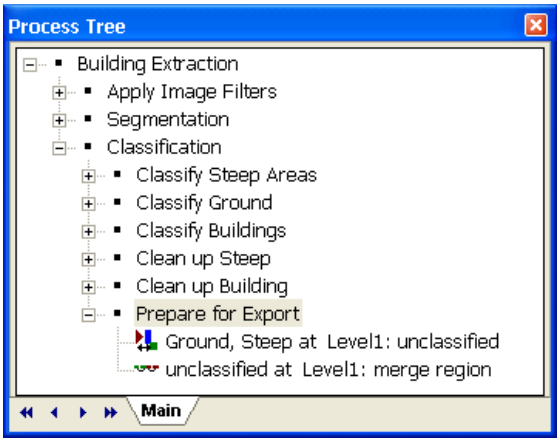
All 'Steep' and 'Ground' objects are now declassified.

11.2.3 Create process to merge all 'unclassified' objects

1. Right-click on the last process and select 'Append New' from the context menu.
2. Select '**merge region**' form the algorithms' list.
3. Define '**unclassified**' as 'Class filter'.
4. **Execute** the process.

All 'unclassified' objects are now merged.

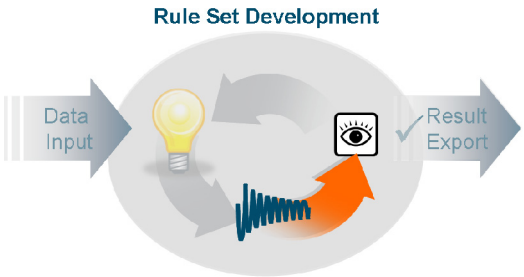




Rule Set Check

Figure 111: Process Tree with parent process and classification process added.

Next step: Review Result!



11.3 Review result

The background to the buildings is now 'unclassified' and merged.

Result Check

11.4 Translating strategy – remove small objects

This chapter covers the following content

➔ *Use the 'Image Object Table' tool to review the smallest objects*

➔ *Use 'remove objects' algorithm to "swallow" small objects*

11.4.1 Use the 'Image Object Table' tool to review the smallest objects

Introduction

As already explained in chapter 10.8.2 the '**Image Object Table**' window allows you to sort the image objects according to feature values and by clicking on an object listed in the table, it will be automatically displayed in the viewer.

Configure the table to list additionally all 'unclassified' objects and their according size

Action!

1. To open the '**Image Object Table**' window do one of the following:
   - Choose **'Image Objects > Image Object Table'** on the main menu bar

   or
   - Choose '**View > Windows > Image Object Table**' on the main menu bar
2. Right-click in the text area of the 'Image Object Table' window and choose '**Configure Image Object Table…'** from the context menu.

The '**Configure Image Object Table'** dialog box opens.

3. Switch on the check-box 'Unclassified image objects' at the lower left.
4. Click **OK** to display all 'Building' and 'unclassified' image objects and their feature values.

Settings Check

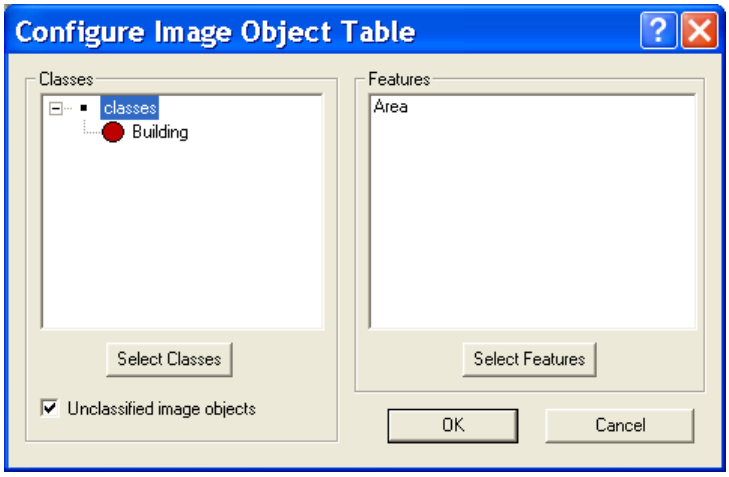


Figure 112: 'Configure Image Object Table' window, with class 'Building' selected and feature 'Area' selected.

Sort the values and browse to the individual objects

Action!

1. Click the 'Area' column header to sort according to the column values. Make the **smallest value** the top of the list.
2. Zoom into the image so that you can see the individual 'unclassified' objects.
3. Click in one row in the table to select the corresponding image object.
4. Evaluate the objects from smaller to larger, objects smaller than about 100 are actually no buildings.

A lot of 'unclassified' objects are still very small.

11.4.2 Use 'remove objects' algorithm to "swallow" small objects

Information

The algorithm 'remove objects' **merges** each image object of the defined domain into the neighbor image object with the **largest common border**. In our case the 'unclassified' objects smaller than **100 pixels** shall to be merged with the neighboring classes.

Insert process to remove small objects

Action!

1. Append a new process and select '**remove objects**' algorithm from the 'Basic Object Reshaping' section.
2. As 'Class filter' set '**unclassified**'.
3. As 'Threshold condition' define 'Area' **smaller than 100** pixel.
4. Keep '**none**' as 'Target class'.

Note:

The field 'Target class' defines the class, who is allowed to incorporate the object defined in the domain. 'None' means here no restrictions to a specific class, all classes are allowed to "swallow" an object.

5. **Execute** the process.

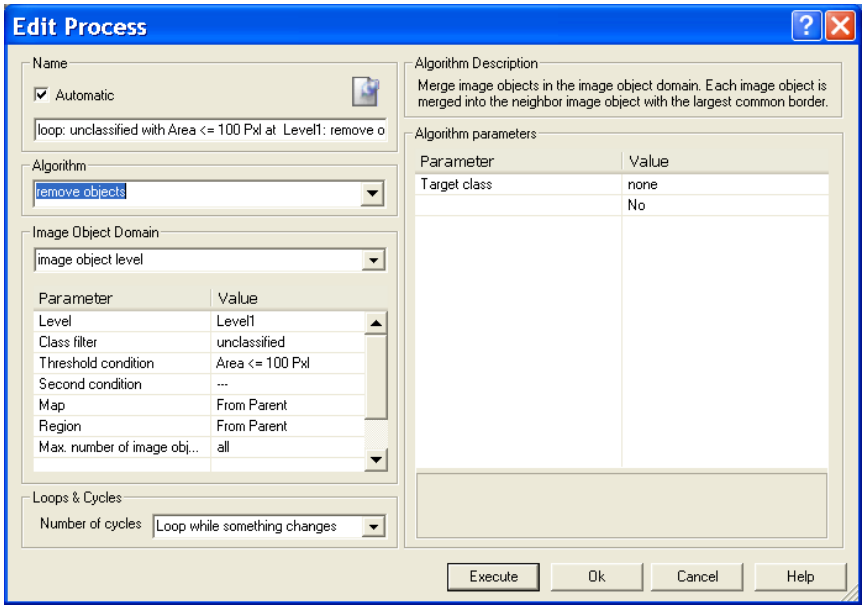


Figure 113: Process settings to remove too small 'unclassified' objects.

Settings Check

Figure 114: Process Tree to prepare the export.

Rule Set Check

11.5 Review result

Check with the 'Image Object Table' the object sizes.

Result Check

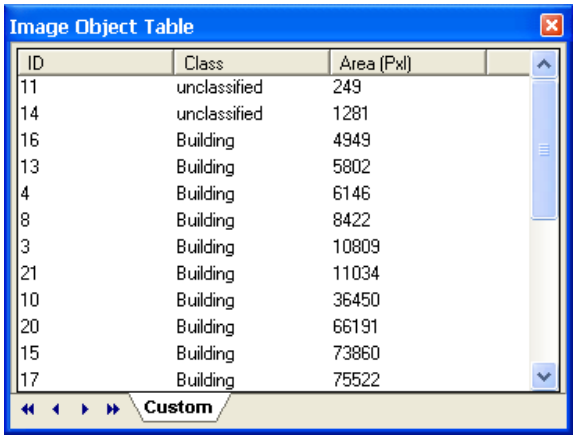


Image Object Table

ID	Class	Area (Pxl)
11	unclassified	249
14	unclassified	1281
16	Building	4949
13	Building	5802
4	Building	6146
8	Building	8422
3	Building	10809
21	Building	11034
10	Building	36450
20	Building	66191
15	Building	73860
17	Building	75522

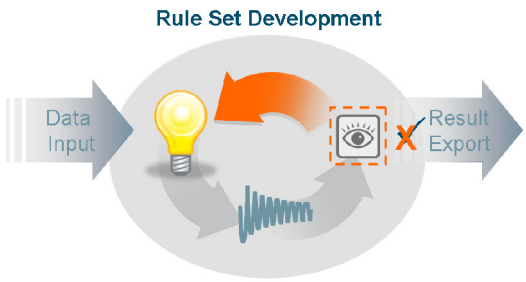
Custom

Figure 115: 'Image Object Table' sorted by 'Area' feature.

Lesson 11 covered the following content

- *Strategy to export the results – declassify those classes not of interest, merge objects*
- *Translating strategy – declassify 'Steep' and 'Ground'*
- *Review result*
- *Translating strategy – remove small objects*
- *Review result*

Next step: Classification accuracy accepted, ready to export result!



Lesson 12 Exporting the vector file

This lesson covers the following content

- *Insert process to export shape file with attributes*
- *Review the exported result*

Introduction

Within eCognition Developer there are several algorithms to export the results. You can choose between exporting

- Statistics (project or object related)
- Images
- Vector files

In this Guided Tour a vector file containing the area of the buildings is exported.

12.1 Insert process to export shape file with attributes

This chapter covers the following content

- *Create the feature 'Class Name'*
- *Insert process to export vector file*

12.1.1 Create the feature 'Class Name'

How to create the feature in detail, please see Guided Tour 1 'Getting started', in Chapter 11.2.1.

Action!

1. In the 'Feature View' window browse to '**Class-Related features>Relations to Classification>Class name**'.
2. Create the feature with keeping the default settings.

The feature has been added to the Feature View list.

Settings Check

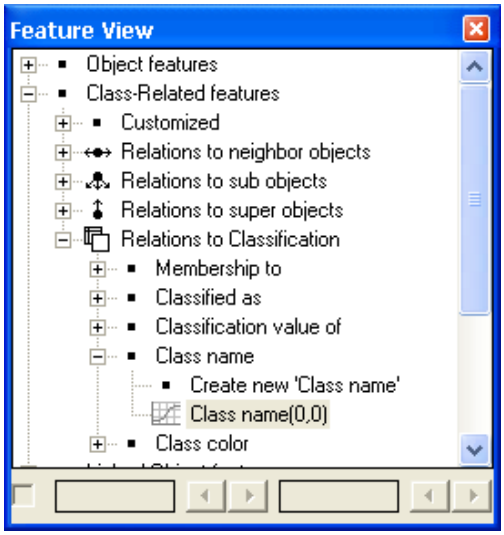


Figure 116: The feature 'Class name' in the Feature View tree.

12.1.2 Insert process to export vector file

Action!

1. In the Process Tree collapse the parent process 'Classification' and right-click on it. Select 'Append New' from the context menu.

Choose algorithm and object domain

2. Choose '**export vector layers**' from the algorithm list. You can find it in the category 'Export'.
3. Keep the '**none**' as 'Class filter'.

Define the parameter

4. In the field 'Export mode' keep 'Use export item'.
5. In the field 'Export item name' insert '**Advanced Building Extraction**'. This will be the name of the exported file.

Define the attributes to be exported

6. Click in the field 'Attribute table'.

The 'Select Multiple Features' dialog box opens.

7. Browse to the 'Class-Related features>Relations to Classification>**Class name**' and double-click on it to shift it to the 'Selected' window.
8. Do the same for the '**Area**' feature (Object features>Geometry>Extent>Area). Confirm with 'OK'.

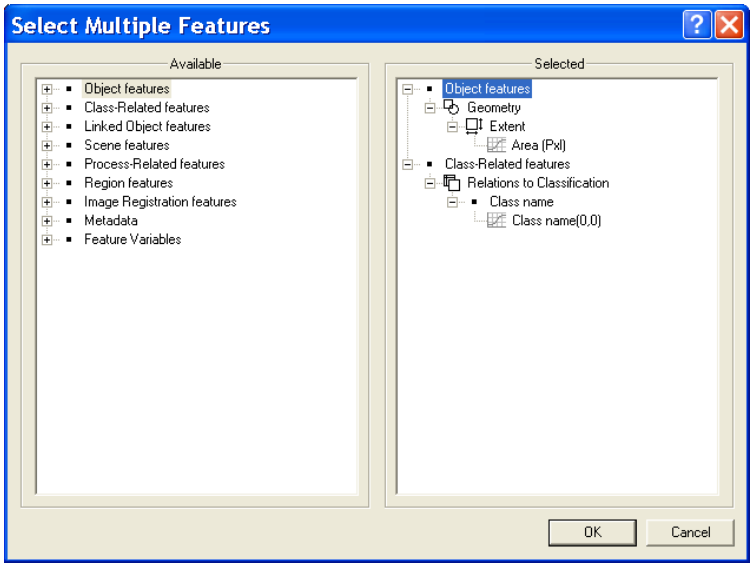


Figure 117: The 'Select Multiple Features' dialog box.

Settings Check

9. Confirm the '**Edit Attribute Table Columns**' dialog box with '**OK**'.
10. In the field '**Shape Type**' change from 'Points' to '**Polygons**'.
11. Keep all other default settings.
12. **Execute** the process.

Action!

The shape file together with its attributes is exported to the location where the image data is stored.

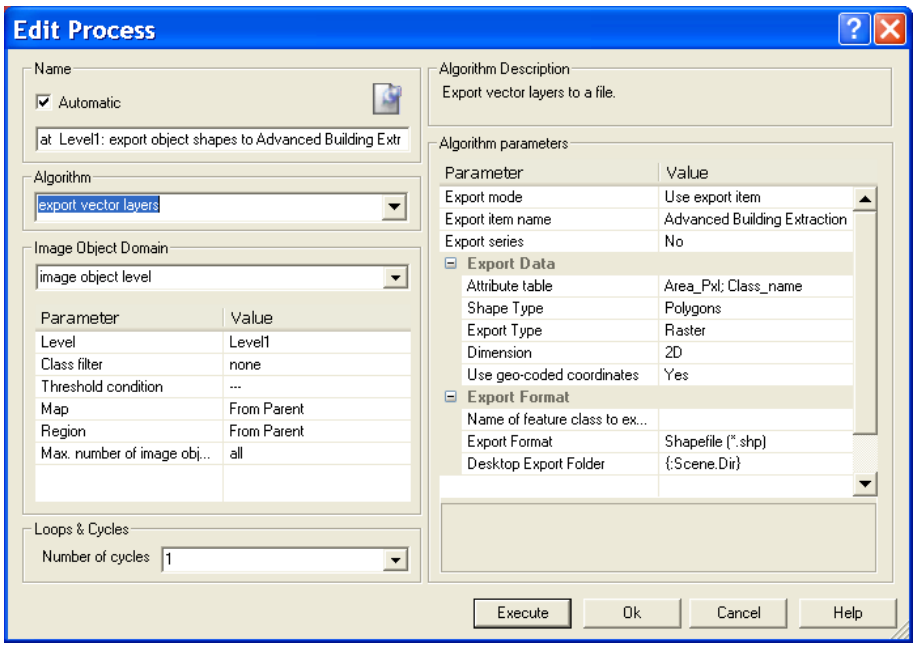


Figure 118: 'Edit Process' dialog box with settings to export a vector layer.

Settings Check

Rule Set Check

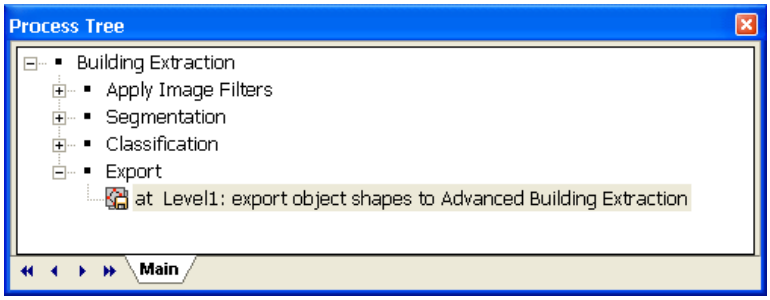


Figure 119: Process Tree with process added to export a vector layer of the result.

12.2 Review the exported result

A shape file has been exported. Review it with ArcGIS or simply check the exported .dbf file.

Result Check

Figure 120: Final classification of the buildings.

	A	B
1	Area_Pxl	Class_name
2	206706	Building
3	151715	Building
4	128403	Building
5	7014	Building
6	10834	Building
7	88103	Building
8	77147	Building
9	157566	Building

Figure 121: Exported .dbf table containing the area and the classification of every object.

Lesson 12 covered the following content

- *Insert process to export shape file with attributes*
- *Review the exported result*

Where to get additional help and information?

eCognition Community and Rule Set Exchange platform

The eCognition community helps to share knowledge and information within the user, partner, academia and developer community to benefit from each other's experience. The community contains next to other content:

- **Wiki**: collection of eCognition related articles (e.g. Rule Set tips and tricks, strategies, algorithm documentation...).
- **Discussions**: ask questions and get answers.
- **File exchange**: share any type of eCognition related code such as Rule Sets, Action Libraries, plug-ins...
- **Blogs**: read and write insights about what's happening around our industry..

Share your knowledge and questions with other users interested in using and developing image intelligence applications for Earth Sciences at http://community.ecognition.com/.

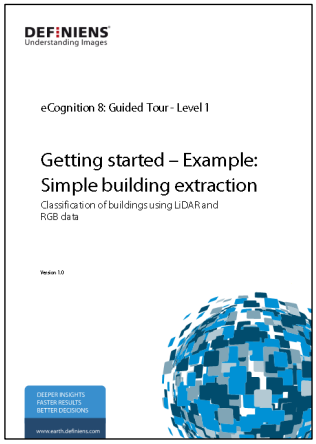
User Guide and Reference Book

Together with the software a User Guide and a Reference book is installed. You can access them in the Developer interface in the main menu 'Help>eCognition Developer User Guide' or Reference Book.

The Reference Book lists detailed information about algorithms and features, and provides general reference information.

Both are also available in the Wiki section of the community http://community.ecognition.com/home/app_wiki

Additional Guided Tours and Tutorials



If you are interested in further auto learning material you can work through the other Guided Tours you can download them on the eCognition community (http://community.ecognition.com/home/app_wiki).

Overview over the current Guided Tours:

- Level1_GuidedTour_LiDAR_SimpleExample
- Level2_GuidedTour_MappingImperviousSurface
- Level3_GuidedTour_LiDAR_AdvancedExample

Tutorials can be found in the Wiki section of the community
http://community.ecognition.com/home/training-material

eCognition Training

eCognition Training Services offer a carefully planned curriculum that provides hands-on, real-world training. We are dedicated to enhancing customers' image analysis skills, helping these organizations to accomplish their goals. Tailored courses are available to meet the needs of customers engaged in Life and Earth Sciences.

Our courses are held in our classrooms around the world and at customer's sites. We offer regular trainings as Open Classes, where anyone can register and as In-Company Training. We also offer Customized Courses to satisfy customer's unique image analysis needs, thereby maximizing the training effect.

For more information please see our website at
http://www.ecognition.com/learn/trainings or write an email to training@definiens.com.

eSeminars

Join one of the live, online eSeminars:

http://www.ecognition.com/learn/trainings/web-based-self-study-training

Consulting

eCognition Consulting Services experts have deep insight into products, best practices and professional project management skills. As a result, we are able to deliver rapid implementations that maximize the return on investment and minimize total cost of ownership. Consulting can also provide full-service, turnkey solutions, including project management, off-site rule development, QA and implementation reviews. Customers can count on our consultants' considerable expertise in the application of technology.

Buy Software and Services?

If you want to purchase software or if you want to know more about pricing please visit http://www.ecognition.com/buy and get in contact with our sales team.