DEFINIENS®
Understanding Images

What's new in eCognition Developer 8?

Working with LiDAR *.las files

New OBIA dimensions

DEEPER INSIGHTS
FASTER RESULTS
BETTER DECISIONS

www.earth.definiens.com

Imprint and Version

Document Version

Published by

Definiens AG
Trappentreustr. 1
D-80339 München
Germany

Phone +49-89-231180-0
Fax +49-89-231180-90

Web http://earth.definiens.com

Legal Notes

Table of Contents

Introduction to this Module

This Module you will learn how to import *.las files, convert and interpolate them so that they can be used for segmentation and classification routines.

This Module has four Lessons:

Lesson 1 Introduction to working with LiDAR .las files in eCognition

Lesson 2 Using the LiDAR converter algorithm

0

Interpolating LiDAR data based on image objects

Lesson 4 Segmenting and classifying with the interpolated image layers

Symbols at the side of the document

The symbols at the side of the document shall guide you through the exercises and help you to identify whether to read something or an action is needed or whether the screenshot is meant to be compared with settings in the software.

Introduction

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

Information

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

Action!

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.

Settings Check

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

Rule Set Check

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

Result Check

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

Lesson 1 Introduction to working with LiDAR .las files in eCognition

This Lesson has the following chapters

Introduction

When dealing with LiDAR processing and analysis, there are several components which provide you with means to directly load and analyze LiDAR point clouds as well as export raster result images such as DSM / DTM.

Working with LiDAR point clouds in eCognition is implemented in a three step approach.

- LiDAR point clouds are loaded and displayed using a quick resampling of the intensity data.
- Once loaded, additional layers can be generated using a special LiDAR converter algorithm.
- Gaps within the LiDAR data can then be interpolated based on Image Objects

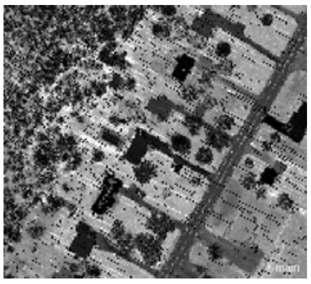
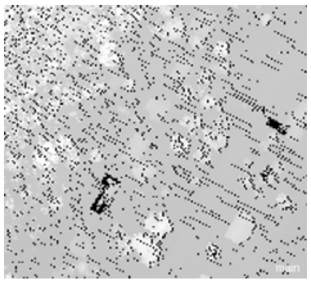
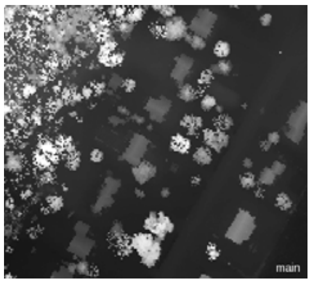
		

Figure 1: LiDAR intensity as displayed after loading.	Figure 2: LiDAR first return from elevation as displayed after applying the LiDAR converter algorithm.	Figure 3: LiDAR last return from elevation as displayed after interpolating.

1.1 Working with LiDAR in this module

Introduction

In this Module you will use elevation information and the information about the number of returns to classify trees and buildings.

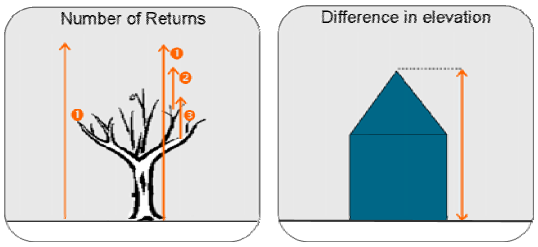


Figure 4: Schematic figure showing the different number of returns.

This information is not represented in the per default loaded LiDAR image layer. Using an algorithm two additional image layers are created, which contain the desired information. These layers are then used for segmentation and classification.

1.2 Loading LiDAR point cloud files in *.las format

This Chapter has the following sub-chapters

→ *Creating a new project using LiDAR *.las data*

→ *Review loaded .las and optical files*

Information

To allow for quick display of the point cloud, a rasterization is implemented in a simple averaging mode based on intensity values. Complex interpolation of data can be done based on the LiDAR file converter algorithm.

In the loading process **a resolution needs to be set** which determines the grid spacing of the raster generated out of the *.las file. The **resolution is set to 1** per default, which is the optimal value for LiDAR data with a point density of **one point per square meter**. For coarser resolution data, set the value to 2 or above, for higher resolution data set it to 0.5or below.

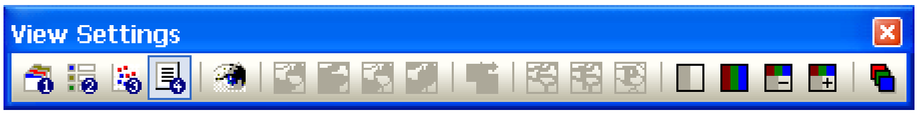
1.2.1 Creating a new project using LiDAR *.las data

Information

A *.las file is imported as any other data via the 'Import Image Layers' dialog box.

Preparation

1. Start Definiens eCogniton Developer in 'Rule Set' mode.
2. Switch to predefined view setting number 4 'Develop rulesets'

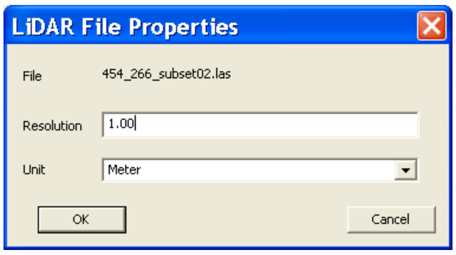


Importing the file

1. In the main menu 'File' choose '**New Project…**' or click on the 'Create New Project' button in the toolbar.
2. Browse to '**454_266_subset02.la**s' which is located in the folder …WhatsNew_eCog8\Data\LiDAR' where the training data is stored.
3. Select the '454_266_subset02.las' to import it.

The 'LiDAR File Properties' dialog box opens.

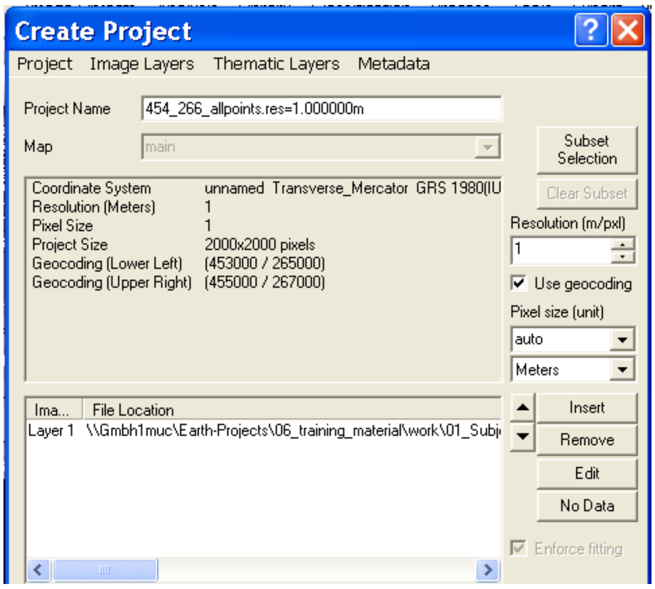
4. In the '**Resolution**' field set **1** and hit '**OK**'.



File 454_266_subset02.las
Resolution 1.00
Unit Meter

Figure 5: 'LiDAR File Properties' dialog box.

The .las file is added to the Project and the Project automatically is named after the imported .las file.



Project Name 454_266_allpoints.res=1.000000m
Map main

Coordinate System	unnamed Transverse_Mercator GRS 1980(IU
Resolution (Meters)	1
Pixel Size	1
Project Size	2000x2000 pixels
Geocoding (Lower Left)	(453000 / 265000)
Geocoding (Upper Right)	(455000 / 267000)

Resolution (m/pxl) 1
Use geocoding ✓
Pixel size (unit) auto
Meters

Ima...	File Location
Layer 1	\\Gmbh1muc\Earth-Projects\06_training_material\work\01_Subj

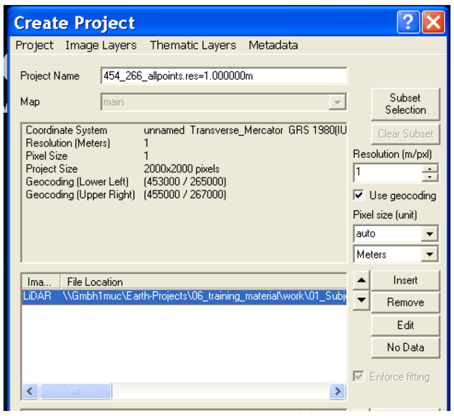
Figure 6: 'Create Project' dialog box with LiDAR data loaded.

5. To finally create the new Project, hit the 'OK' button of the 'Create Project' dialog box.

1.2.2 Review loaded .las and optical files

Information

LiDAR point clouds are loaded and displayed using a quick resampling of the intensity data.



Project Name: 454_266_allpoints.res=1.000000m
Map: main

Coordinate System	unnamed Transverse_Mercator GRS 1980(IU
Resolution (Meters)	1
Pixel Size	1
Project Size	2000x2000 pixels
Geocoding (Lower Left)	(453000 / 265000)
Geocoding (Upper Right)	(455000 / 267000)

Resolution (m/pxl): 1
[✓] Use geocoding
Pixel size (unit): auto, Meters

Ima...	File Location
LiDAR	\\Gmbh1muc\Earth-Projects\06_training_material\work\01_Subj

Figure 7: Project with loaded LiDAR data

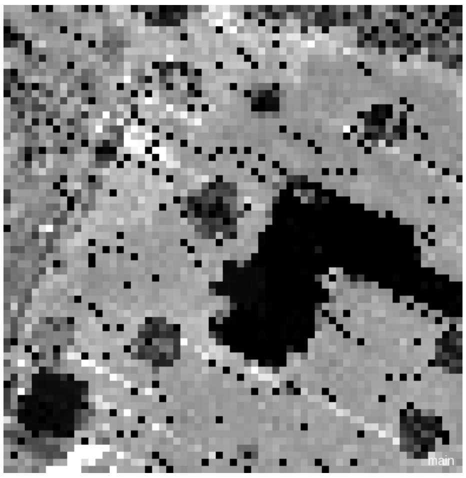


Figure 8: Detail of loaded LiDAR data.

Lesson 2 Using the LiDAR converter algorithm

This Lesson has the following chapters

- → *Introduction to algorithm 'LiDAR File Converter'*
- → *Creating image layer 'Elevation First' using algor. 'LiDAR' converter*
- → *Creating image layer 'Number of Returns' using algor. 'LiDAR' converter*

Introduction

In this and the following Lessons a project will be used, which contains LiDAR data in combination with optical data.

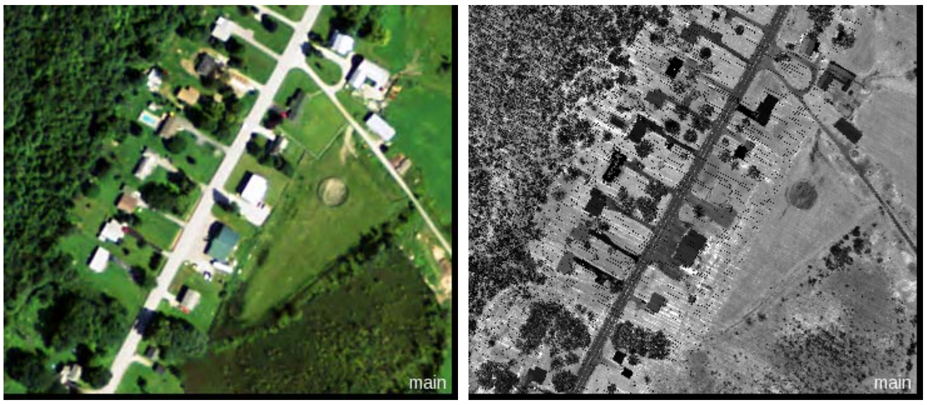
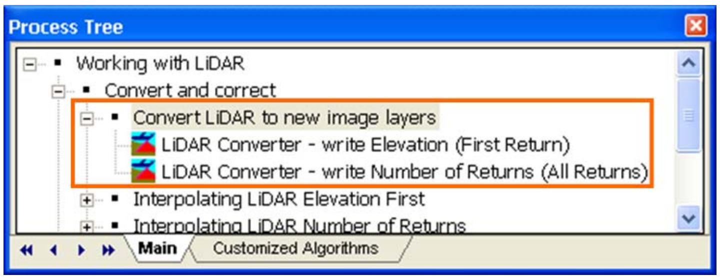


Figure 9: Left: Optical image layers; Right: loaded LiDAR data.

From the loaded LiDAR data, additional information can be extracted using the new algorithm '**LiDAR File Converter**'. This algorithm can create **new image layers**, containing information about elevation, number of returns and more.



Process Tree

- Working with LiDAR
  - Convert and correct
    - Convert LiDAR to new image layers
      - LiDAR Converter - write Elevation (First Return)
      - LiDAR Converter - write Number of Returns (All Returns)
    - Interpolating LiDAR Elevation First
    - Interpolating LiDAR Number of Returns

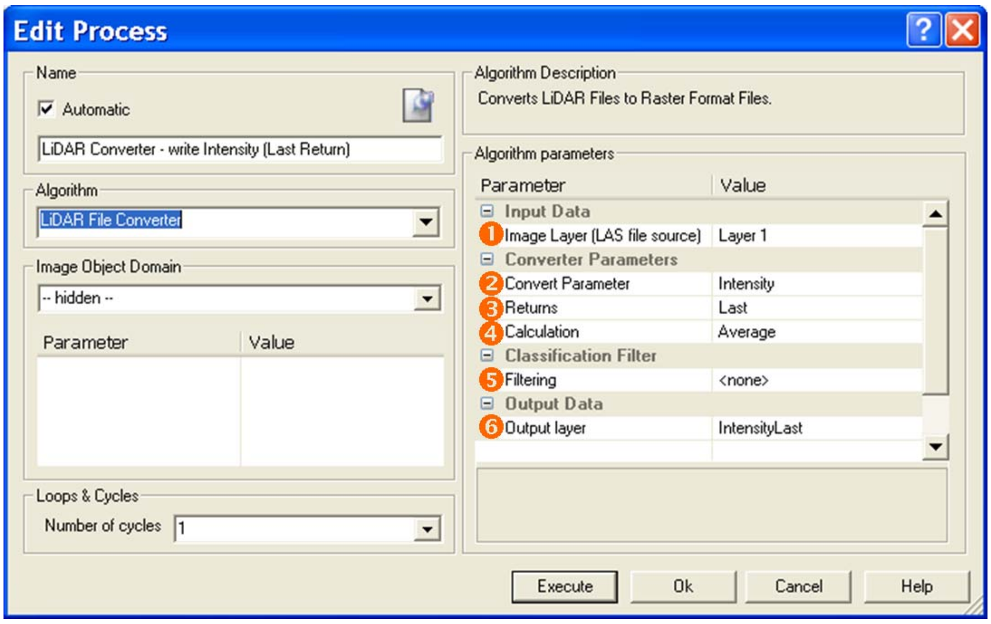
Figure 10: Process Tree with Processes highlighted to convert LiDAR data.

In the later exercises the image layers created in this Lesson are used for further **segmentation** and **classification** of trees and buildings in the loaded subset. The number of returns is used to classify trees, the elevation is used to classify buildings.

Introduction

2.1 Introduction to algorithm 'LiDAR File Converter'

This algorithm can create **new image layers**, containing information about elevation, number of returns and more.



Name
☑ Automatic
LiDAR Converter - write Intensity (Last Return)

Algorithm
LiDAR File Converter

Image Object Domain
-- hidden --

Loops & Cycles
Number of cycles 1

Algorithm Description
Converts LiDAR Files to Raster Format Files.

Algorithm parameters

Parameter	Value
Input Data	
1 Image Layer (LAS file source)	Layer 1
Converter Parameters	
2 Convert Parameter	Intensity
3 Returns	Last
4 Calculation	Average
Classification Filter	
5 Filtering	<none>
Output Data	
6 Output layer	IntensityLast

Figure 11: Process settings to convert the LiDAR to an new image Layer.

Image Layer (Las file source)

In the field '**Image Layer (LAS source file)**' 1, it is defined which layer to be used fro converting, respectively the dataset from which to extract additional layers.

Convert Parameter

In the field '**Convert Parameter**' 2 it can be defined which information to use for generating new image layers. The following parameters can be set:

Convert Parameter:	**Operation:**
Intensity	The point intensity values are used to generate the raster layer.
Elevation	The point elevation values are used to generate the raster layer.
Class	The class information stored in the points is used to generate the raster layer.
Number of returns	A raster layer representing the number of returns is generated.

Returns

Since the data is resampled into a 2D Grid with a fixed spacing during import in the software, several LiDAR points can be within one pixel.

In the field '**Returns**' 3 it can be defined which points to use in generating the raster. The following settings are available.

Return type:	**Operation:**
First	Only first return points are used
Last	Only last return points are used
All	All points are used

Calculation

In the field '**Calculation**' 4 the calculation mode can be defined, which allows determining how to compute the resulting pixel value out of several LiDAR points. The following modes are available:

Calculation modes:	**Calculation:**
Average	The average out of all point values is calculated
Minimum	The minimum value out of all available point values is used
Maximum	The maximum value out of all available point values is used
Median (lower)	The lower median out of all available point values is used
Median (upper)	The upper median out of all available point values is used
Most frequent value	The most frequent value out of all available point values is used

Filtering

In the field '**Filtering**' 5 different filtering modes allow to limit the points used for generating the raster representation based on available point properties. The following options are available:

Filtering mode:	**Operation:**
No filtering	No filtering is applied. This is the default setting
Selected classes	All operations are applied to points of the selected classes only
Non selected classes	All operations are applied to points of the classes which are not selected only

If filtering is applied, a list of classes as standardized for the *.las file format is provided. Standard classes allow using predefined standardized class types, for custom classes the other classes selection is available.

Output Layer

In the field '**Output Layer**' 6 an individual name for the layer to be created can be defined.

2.2 Creating image layer 'Elevation First' using algor. 'LiDAR' converter

Information

This Chapter has the following sub-chapters

- → *Preparation*
- → *Create the image layer*
- → *Execute and review the created image layer 'Elevation First'*

In the example Project, the first converted layer shall contain the value of the laser point with the maximum value of first return for elevation. This layer will be later used to classify buildings.

2.2.1 Preparation

Action!

1. In the main menu 'File' choose '**Open Project**' or click on the 'Open Project' button in the toolbar.
2. Open the project '**Working with LiDAR las files.dpr**' in the folder '…\WhatsNew_eCog8\Projects\LiDAR' at the location where the training data is stored.
3. View the image layer 'LiDAR' and view the optical data.

The loaded Project has optical data and an imported LiDAR .las file. From the .las file the intensity is shown in the image layer 'LiDAR'. A Process Tree is loaded, nothing is executed.

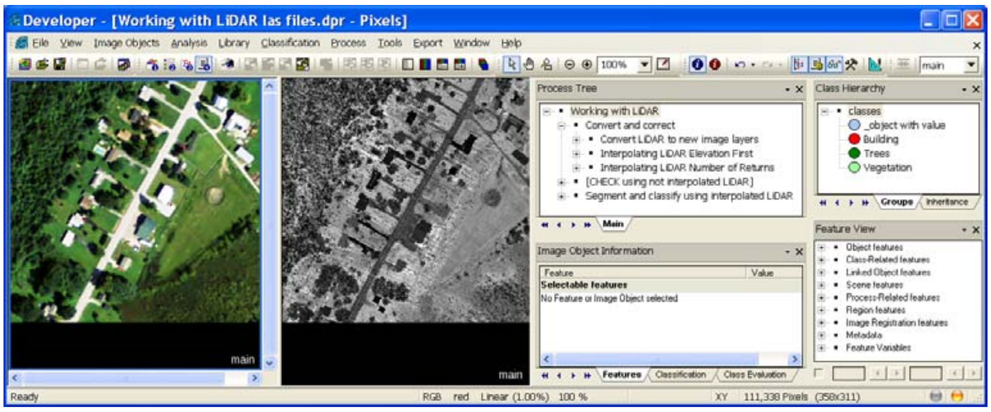

Developer - [Working with LiDAR las files.dpr - Pixels]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
Process Tree
Working with LiDAR
Convert and correct
Convert LiDAR to new image layers
Interpolating LiDAR Elevation First
Interpolating LiDAR Number of Returns
[CHECK using not interpolated LiDAR]
Segment and classify using interpolated LiDAR
Main
Class Hierarchy
classes
_object with value
Building
Trees
Vegetation
Groups
Inheritance
Image Object Information
Feature
Value
Selectable features
No Feature or Image Object selected
Features
Classification
Class Evaluation
Feature View
Object features
Class-Related features
Linked Object features
Scene features
Process-Related features
Region features
Image Registration features
Metadata
Feature Variables
main
Ready
RGB red Linear (1.00%) 100 %
XY 111,338 Pixels (358x311)

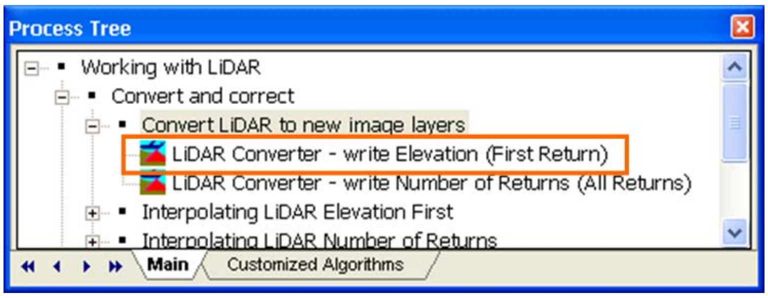
Figure 12: The loaded Project 'Working with LiDAR las files.dpr'.

2.2.2 Create the image layer

1. In the Process Tree **expand** the Parent Processes 'Working with LiDAR', 'Convert and correct' and '**Convert LiDAR to new image layers**'.

2. Double-click on the Process '**LiDAR Converter - write Elevation (First Return)**' to open it.

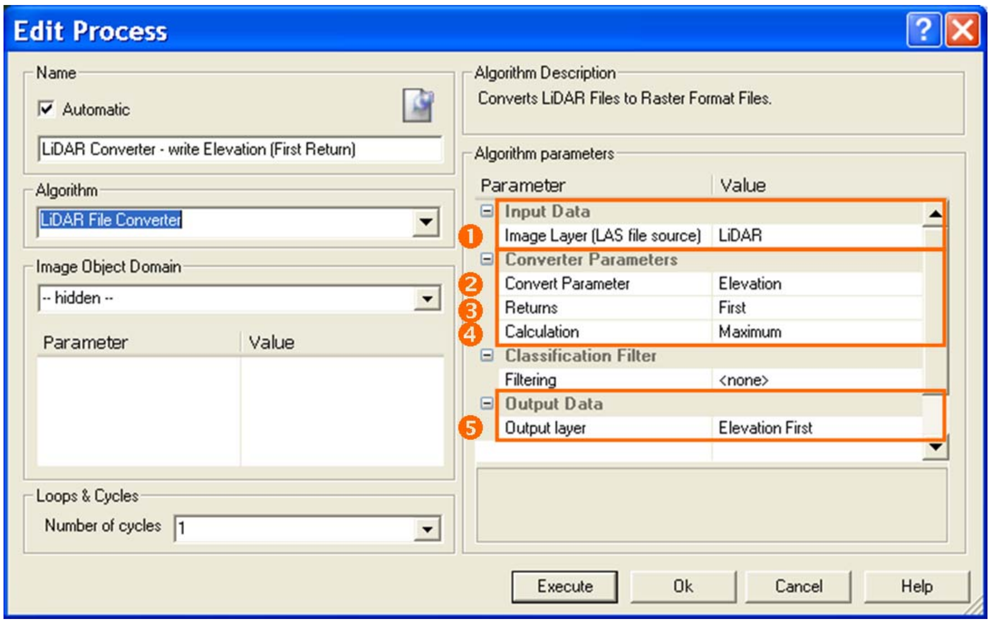
Action!



Process Tree

- Working with LiDAR
  - Convert and correct
    - Convert LiDAR to new image layers
      - LiDAR Converter - write Elevation (First Return)
      - LiDAR Converter - write Number of Returns (All Returns)
    - Interpolating LiDAR Elevation First
    - Interpolating LiDAR Number of Returns

Figure 13: Process Tree with Process highlighted to convert to 'Elevation First Return'.



Edit Process

Name: ☑ Automatic — LiDAR Converter - write Elevation (First Return)

Algorithm: LiDAR File Converter

Image Object Domain: -- hidden --

Loops & Cycles — Number of cycles: 1

Algorithm Description: Converts LiDAR Files to Raster Format Files.

Algorithm parameters

Parameter	Value
Input Data	
Image Layer (LAS file source)	LiDAR
Converter Parameters	
Convert Parameter	Elevation
Returns	First
Calculation	Maximum
Classification Filter	
Filtering	<none>
Output Data	
Output layer	Elevation First

Settings Check

Figure 14: Process settings to convert the LiDAR to 'Elevation First'.

- The selected algorithm 'LiDAR File Converter' can be found in the 'LiDAR tools' section of the algorithms list.

Algorithm Parameters

- In the field '**Image Layer (LAS source file)**' ①, 'LiDAR' is chosen from the list and defined as the sources of the .las file.
- In the field '**Convert Parameter**' ❷ the mode 'Elevation' is chosen from the drop down list.
- In the field '**Returns**' ③ 'First' is chosen from the drop-down list.
- In the field '**Calculation**' ④ the calculation mode 'Maximum' is chosen.
- In the field '**Output Layer**' ⑥ the name 'Elevation First' is defined. This will be the name for the new created image layer.

2.2.3 Execute and review the created image layer 'Elevation First'

Action!

1. **Execute** the Process.
2. Display in one Viewer the original '**LiDAR**' image file, in a second Viewer display the new created '**Elevation First**' layer.
3. Hover with the mouse over the new layer 'Elevation First' and evaluate the different values of this layer.

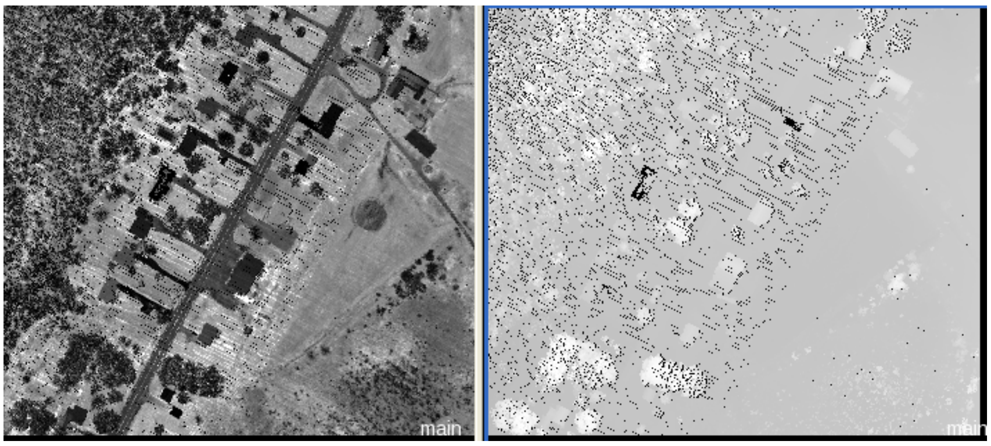


Figure 15: Left: Initial LiDAR data; Right: Converted image layer 'Elevation First'.

2.3 Creating image layer 'Number of Returns' using algor. 'LiDAR' converter

This Chapter has the following sub-chapters

➔ *The Process settings*

➔ *Execute and review the created image layer 'Elevation First'*

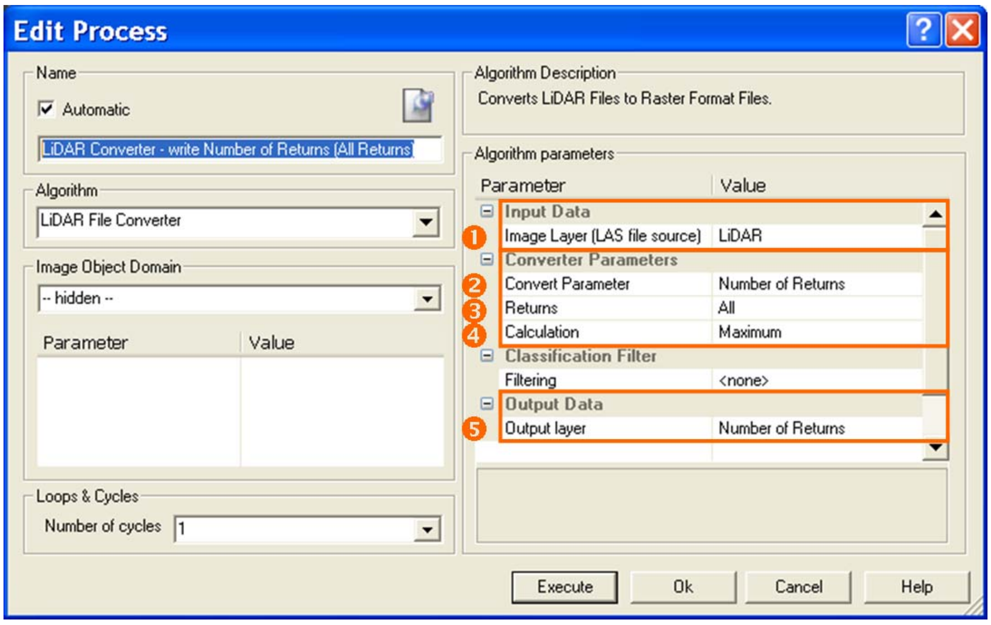
Information

In the example Project, the second converted layer shall contain the value of the laser point with the maximum number of all returns. This layer will be later used to classify trees.

2.3.1 The Process settings

Settings Check

1. Double-click on the Process '**LiDAR Converter - write Number of Returns (All Returns)**' to open it.



Name
- [✓] Automatic
- LiDAR Converter - write Number of Returns (All Returns)

Algorithm: LiDAR File Converter

Image Object Domain: -- hidden --

Parameter	Value

Loops & Cycles
Number of cycles: 1

Algorithm Description
Converts LiDAR Files to Raster Format Files.

Algorithm parameters

Parameter	Value
Input Data	
1 Image Layer (LAS file source)	LiDAR
Converter Parameters	
2 Convert Parameter	Number of Returns
3 Returns	All
4 Calculation	Maximum
Classification Filter	
Filtering	<none>
Output Data	
5 Output layer	Number of Returns

Figure 16: Process settings to convert the LiDAR to 'Number of Returns'.

Algorithm Parameters

- In the field '**Image Layer (LAS source file)**' 1, again '**LiDAR**' is chosen from the list and defined as the sources of the .las file.
- In the field '**Convert Parameter**' 2 the mode '**Number of Returns**' is chosen from the drop down list.
- In the field '**Returns**' 3 '**All**' is chosen from the drop-down list.
- In the field '**Calculation**' 4 the calculation mode '**Maximum**' is chosen.
- In the field '**Output Layer**' 6 the name '**Number of Returns**' is defined. This will be the name for the new created image layer.

2.3.2 Execute and review the created image layer 'Elevation First'

Action!

2. **Execute** the Process
3. **Display** in one Viewer the original 'LiDAR' image file, in a second Viewer display the new created 'Number of Returns' layer.
4. Hover with the **mouse** over the new layer 'Number of Returns' and evaluate the different values of this layer.

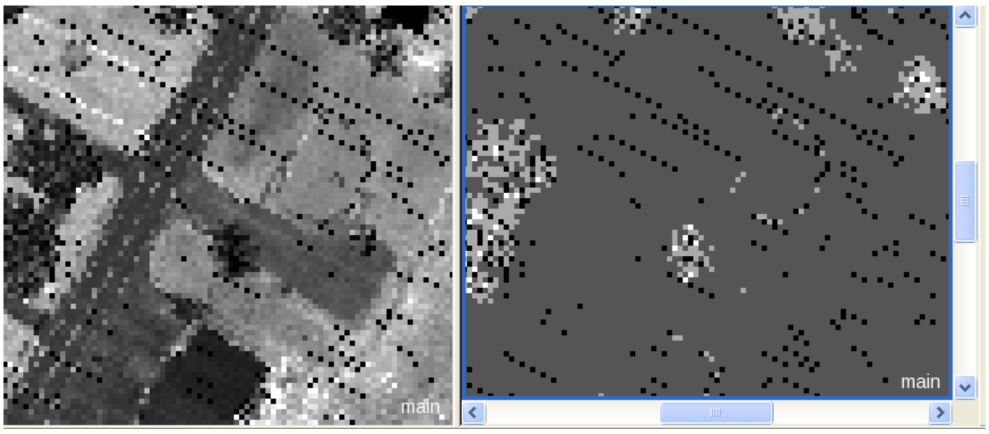


Figure 17: Left: initial LiDAR data; Right: Converted image layer 'Elevation First'.

Lesson 3 Interpolating LiDAR data based on image objects

This Lesson has the following chapters

Introduction

The imported and converted LiDAR images shall be used to help **segmenting** and **classifying** the trees and the buildings in this subset.

Trees are expected to have multiple returns and buildings are expected to have a high difference in elevation to the surrounding neighbors.

Before using the image layers they must be **interpolated**, because there are pixels with **0 values**, disturbing the object creation and feature calculation.

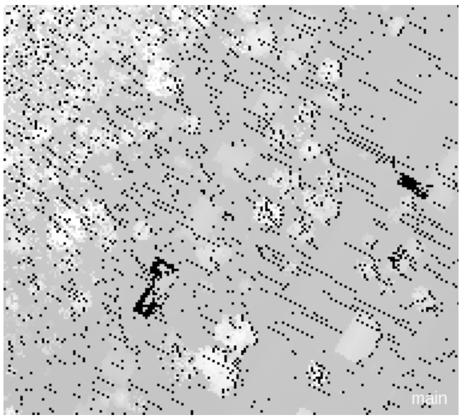


Figure 18: LiDAR first return from elevation as displayed after applying the LiDAR converter algorithm.

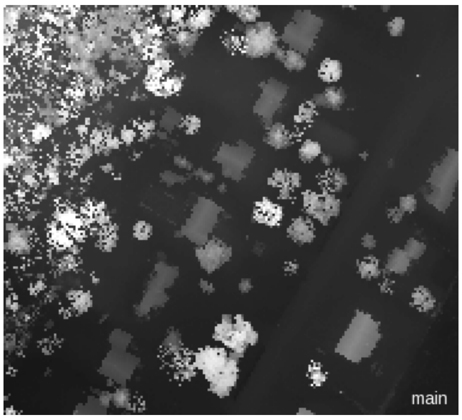


Figure 19: LiDAR last return from elevation as displayed after interpolating.

3.1 Why interpolating?

This Chapter has the following sub-chapters

- → *Segment with not interpolated image layers*
- → *Classify trees with not interpolated image layers*
- → *Classify buildings with not interpolated image layers*

Introduction

To outline the objects best, both LiDAR image layers are used for segmentation. But to **create meaningful** Objects, the converted layers have to be interpolated. Also if later calculating the **mean difference in elevation** for buildings, a 0 value Object would give back misleading values.

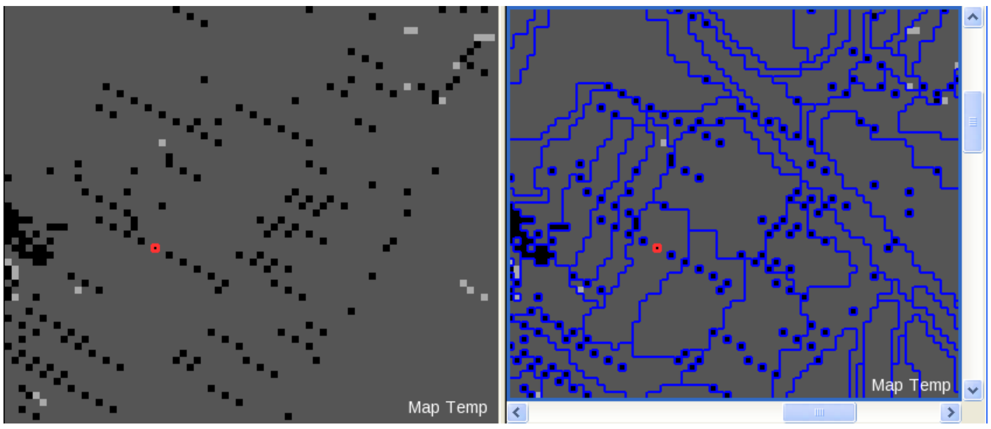


Figure 20: Left: Not interpolated image layer; Right: Objects created using this layer.

Information

3.1.1 Segment with not interpolated image layers

All optical layers are used for segmentation. The two converted LiDAR images are used too, the weighting is set to 10 to balance the lower value range of these layers compared to the optical values.

Action!

The processing without interpolated layers is done on a separate Map.

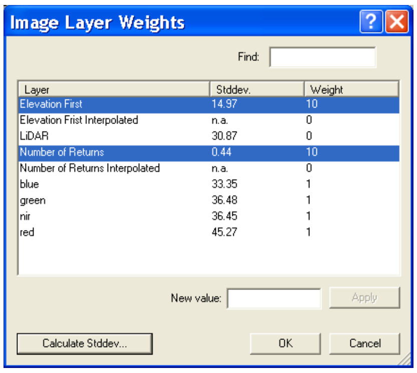
Preparation

1. In the Process Tree **collapse** the Parent Processes 'Convert and correct'.
2. **Expand** the Parent Processes 'CHECK using not interpolated LiDAR'.
3. Execute the Process '**copy map to 'Map Temp''**.

4. Switch in the viewer to the 'Map Temp'.

Process settings

- Both converted **LiDAR layers** are used for segmentation and weighted with 10.



Layer	Stddev.	Weight
Elevation First	14.97	10
Elevation Frist Interpolated	n.a.	0
LiDAR	30.87	0
Number of Returns	0.44	10
Number of Returns Interpolated	n.a.	0
blue	33.35	1
green	36.48	1
nir	36.45	1
red	45.27	1

Figure 21: 'Image Layer Weights' dialog box with both converted image layers selected.

Settings Check

Execute and review the Result

1. **Execute** the Process
2. Switch on the outlines and review the created Objects.

Action!

Where the image layer have 0 values, small Objects are created.

Result Check

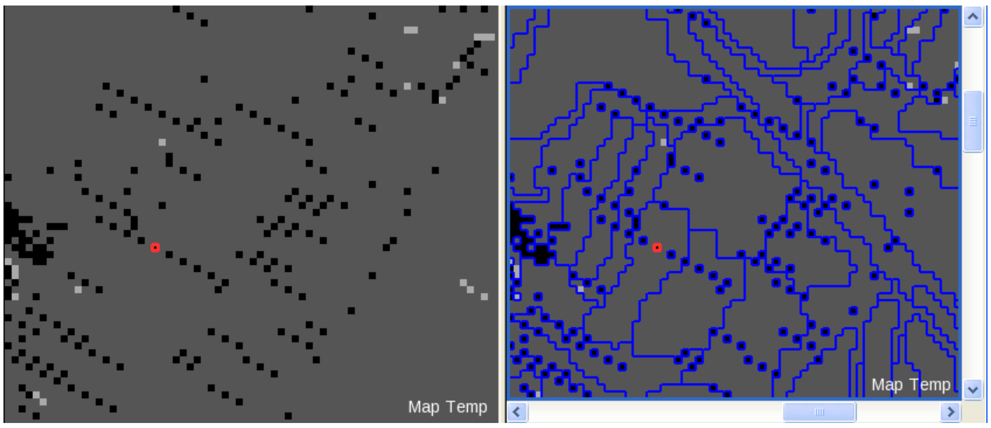


Figure 22: Left: Not interpolated image layer; Right: Objects created using this layer.

3.1.2 Classify trees with not interpolated image layers

The mean value of the image layer 'Number of Returns' is used to classify Trees. In areas where the values of the image layers is 0, no Trees are classified, a lot of holes are in the classification.

Information

Action!

Execute and review the Result

1. **Execute** the Process 'with NDVI > 0 at L2: Vegetation' and 'Vegetation with Mean Number of Returns > 1.1 at L2: Trees'.
2. Switch on the classification view and review the classification.

Result Check

The classification of 'Trees' has a lot of holes.

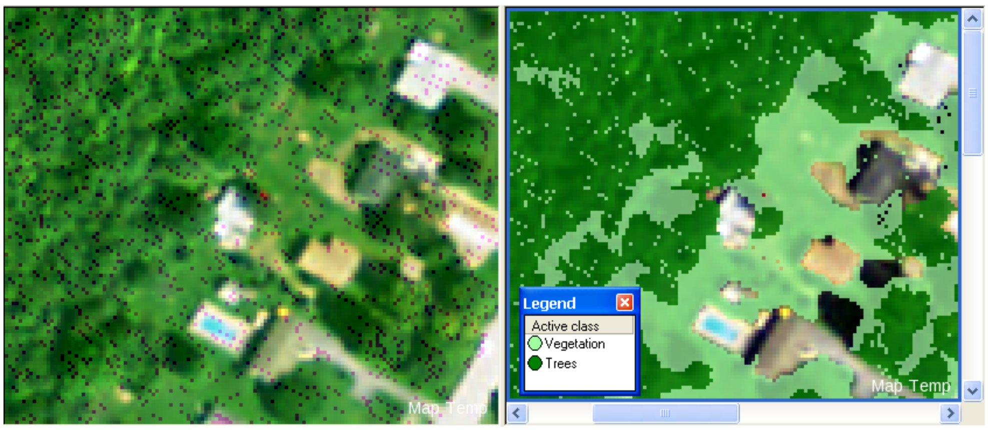


Figure 23:Left: Optical layers and Number of Returns in mixed view; Right: Classification Result.

3.1.3 Classify buildings with not interpolated image layers

Information

Objects with difference in elevation in comparison to their neighbors are classified as 'Building'. In areas where the values of the image layers is 0, the 'Mean difference' calculation gives back values not fitting for 'Building' objects, respectively other objects actually not being a building fulfill the condition. A lot of misclassifications are the result.

Action!

Execute and review the Result

1. **Execute** the Process 'loop: unclassified with MeanDiff_Mean Elevated First_10> 0.5 and Compactness < 2 at L2: Building'.
2. Switch on the classification view and review the classification.

Result Check

A lot of Objects are misclassified as 'Building'.

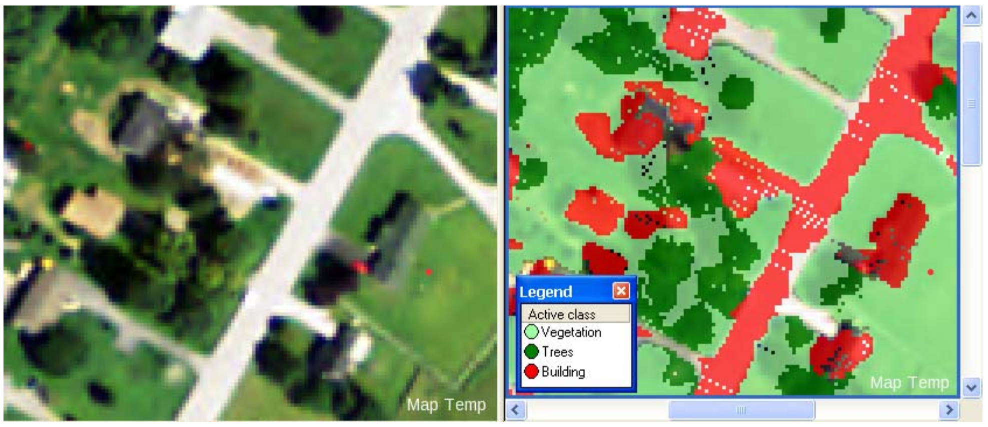


Figure 24: Left: Optical layers; Right: Classification Result.

3.2 Introduction to interpolating LiDAR data based on image objects

Introduction

To avoid problems in Object creation and feature calculation as explained in the chapter before, for both converted LiDAR image layers the 0 value pixels will be interpolated. To do so an object oriented approach is used, as result an interpolated image layer will be added to the Project.

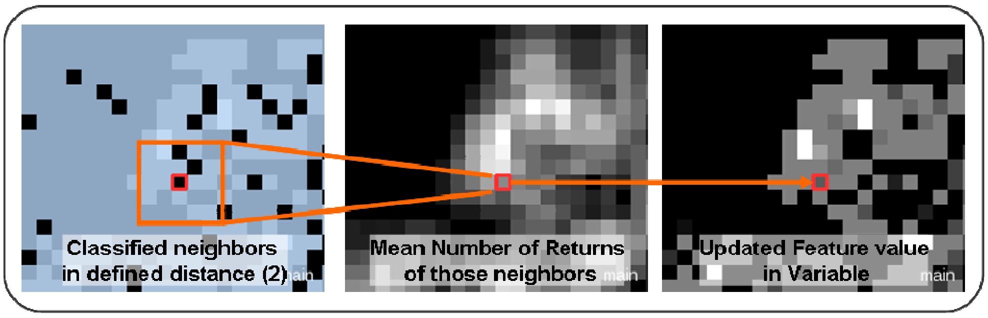


Figure 25: The 'Mean of neighbors' is basis for updating the Objects with 0 values.

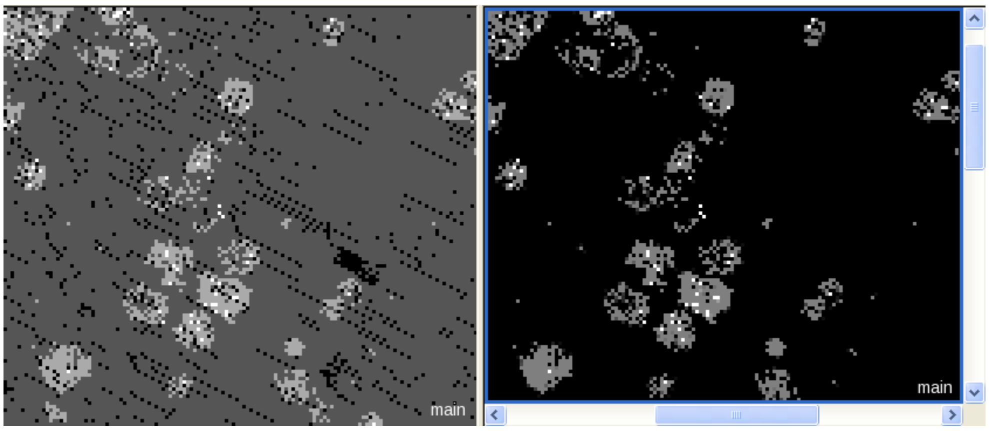


Figure 26: Left: not interpolated image layer; Right: interpolated image layer.

3.3 Storing current feature values in an Object Variable

This Chapter has the following sub-chapters

Information

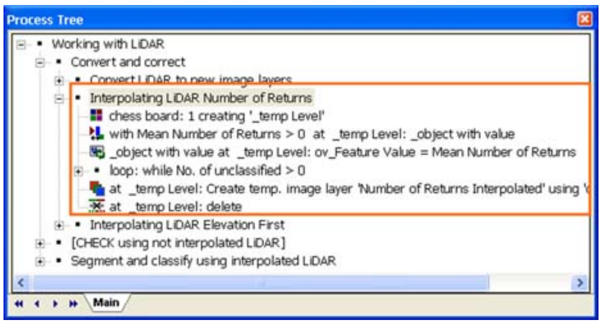
3.3.1 Classifying those Objects with value

First step of this Process sequence is to classify those objects which have a value for the 'Number of Returns' image layer. Therefore a pixel-sized Level is created and those Objects are classified. Later a Customize Class-Related Feature is pointing to these classified Objects.

Action!

1. Switch in the viewer to the **main Map**.
2. Collapse the Parent Process 'CHECK using not interpolated LiDAR'.
3. Expand the Process 'Convert and correct' and '**Interpolating LiDAR Number of Returns**'.

Rule Set Check



```
Process Tree
⊟ • Working with LiDAR
  ⊟ • Convert and correct
    ⊞ • Convert LiDAR to new image layers
    ⊟ • Interpolating LiDAR Number of Returns
        chess board: 1 creating '_temp Level'
        with Mean Number of Returns > 0  at  _temp Level: _object with value
        _object with value at  _temp Level: ov_Feature Value = Mean Number of Returns
      ⊞ • loop: while No. of unclassified > 0
        at  _temp Level: Create temp. image layer 'Number of Returns Interpolated' using '
        at  _temp Level: delete
    ⊞ • Interpolating LiDAR Elevation First
  ⊞ • [CHECK using not interpolated LiDAR]
  ⊞ • Segment and classify using interpolated LiDAR
Main
```

Figure 27: Process Tree with Processes highlighted to interpolate 'Number of Returns'.

Action!

Create a pixel-sized Level and classify

1. Execute the Process 'chess board: 1 creating '_temp Level''.
2. Execute the Process 'with Mean Number of Returns > 0 at _temp Level: _object with value'.

Result Check

Review result

The Objects with 0 value stay unclassified, the others are classified.

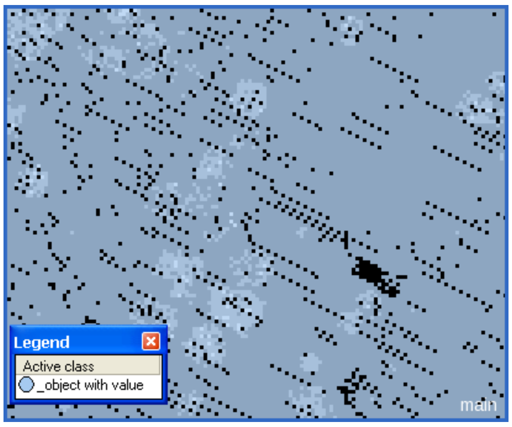


Figure 28: Classification View.

3.3.2 Storing current feature values in an Object Variable

For all Objects which have values, the mean values of 'Number of Returns' is written in an **Object Variable**. This Object Variable is later used to do the interpolation of those Objects with currently no values for 'Number of Returns' layer.

Information

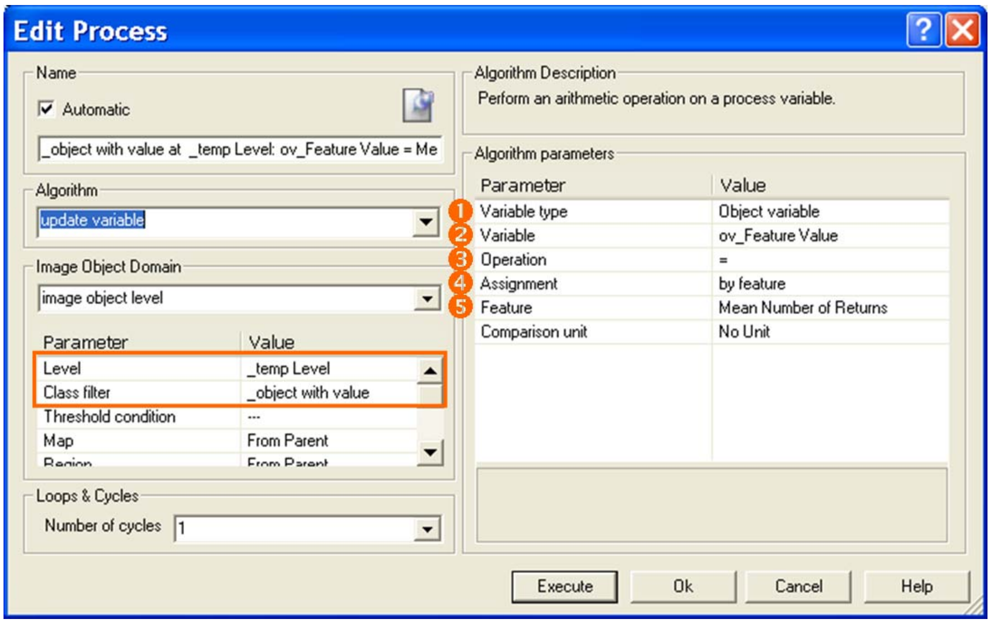
Process settings

1. Double-click on the Process '**_object with value at _temp Level: ov_Feature Value = Mean Number of Returns**'.

Action!

✓

Settings Check



Edit Process

Name: ✓ Automatic

_object with value at _temp Level: ov_Feature Value = Me

Algorithm: update variable

Image Object Domain: image object level

Parameter	Value
Level	_temp Level
Class filter	_object with value
Threshold condition	---
Map	From Parent
Region	From Parent

Loops & Cycles: Number of cycles 1

Algorithm Description: Perform an arithmetic operation on a process variable.

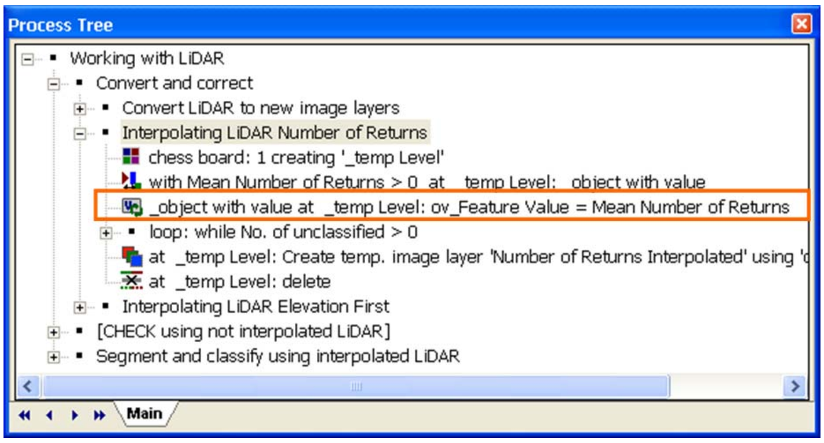
Algorithm parameters

Parameter	Value
1 Variable type	Object variable
2 Variable	ov_Feature Value
3 Operation	=
4 Assignment	by feature
5 Feature	Mean Number of Returns
Comparison unit	No Unit

Figure 29: Process settings to update an Object Variable with values from Feature 'Mean Number of Returns':

- In the Image Object Domain, '**_objects with value**' is defined as Class.
- As 'Variable Type' '**Object variable**' is defined. This variable type assigns a value to every Object.
- In the field 'Variable' the name of the variable is defined, here '**ov_Feature Value**'.
- In the 'Operations' field = is defined.
- In the field 'Assignment' '**by feature**' is defined. The value of the specified feature is written in the Objects specified in the Image Object Domain.
- In the field 'Feature' the feature '**Mean Number of Returns**' is defined.

Rule Set Check



```
Process Tree
- Working with LiDAR
  - Convert and correct
    + Convert LiDAR to new image layers
    - Interpolating LiDAR Number of Returns
        chess board: 1 creating '_temp Level'
        with Mean Number of Returns > 0  at  _temp Level:  _object with value
        _object with value at  _temp Level: ov_Feature Value = Mean Number of Returns
      + loop: while No. of unclassified > 0
        at  _temp Level: Create temp. image layer 'Number of Returns Interpolated' using '
        at  _temp Level: delete
    + Interpolating LiDAR Elevation First
  + [CHECK using not interpolated LiDAR]
  + Segment and classify using interpolated LiDAR
```

Figure 30: Process Tree with Process highlighted to write values in an Object Variable.

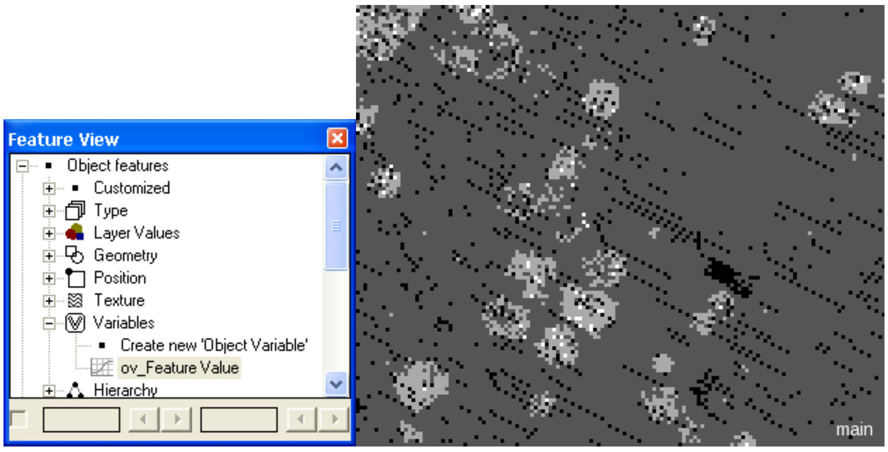
3.3.3 Review the created Object Variable and its values

Action!

2. Execute the Process.
3. Go to the 'Feature View' window and browse to '**Object Features>Variables**'
4. Double-click on '**ov_Feature Value**'.

The Object Variable and the Feature 'Mean Elevation First' have currently the same values.

Result Check



```
Feature View
- Object features
  + Customized
  + Type
  + Layer Values
  + Geometry
  + Position
  + Texture
  - Variables
      Create new 'Object Variable'
      ov_Feature Value
  + Hierarchy
```

Figure 31: Object Variable in the Feature View and its values, currently representing exactly the 'Elevation First' layer.

3.4 Writing new values for those Objects without a value

This Chapter has the following sub-chapters

- → *The Customized Relational Feature to interpolate*
- → *Execute the interpolation*

Information

The processing sequence to write interpolated values for Objects with no value is repeated until no unclassified Object remains.

With in the loop, for all unclassified objects, the value of a **Customized Relational feature** is written in the object variable. The features calculates the mean of the neighboring objects. This value is written in the Object Variable of the unclassified Objects.

The unclassified Object which have now values for the Object Variable are then classified as '_object with value' too.

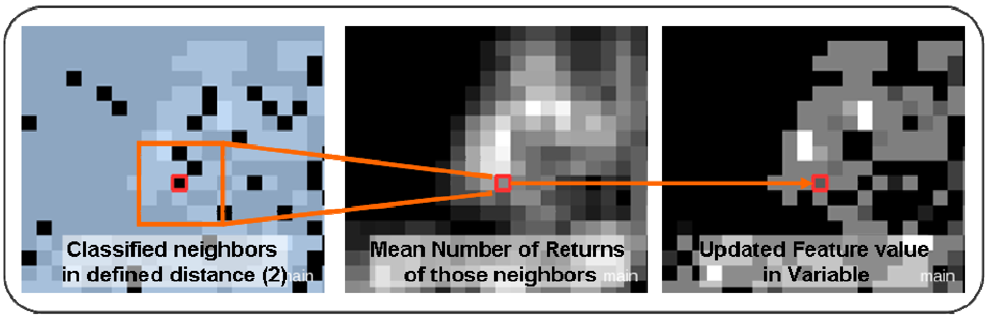


Figure 32: The 'Mean of neighbors' is basis for updating the Objects with 0 values.

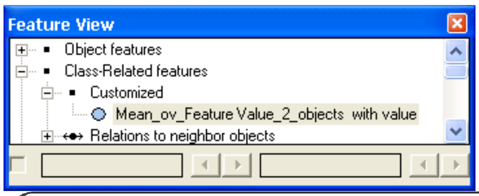
3.4.1 The Customized Relational Feature to interpolate

Information

To calculate the interpolation a **Customized Class-Related Feature** is used, which computes the mean of the **mean values of all neighboring** '_objects with value' Objects in a certain distance. This value is then stored in the Object Variable for those Objects, which have currently 0 value.

Action!

1. Go to the 'Feature View' window and browse to '**Class-Related Features>Customized**'.
2. Double-click on '**ov_Feature Value**'.


Feature View
- Object features
- Class-Related features
  - Customized
    - Mean_ov_Feature Value_2_objects with value
  - Relations to neighbor objects

✓

Result Check

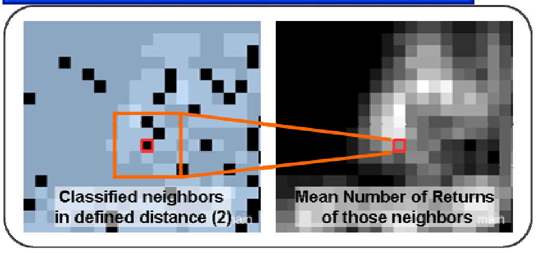


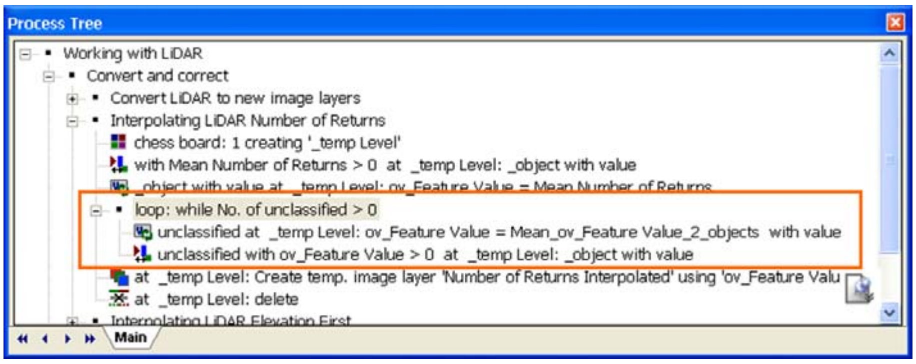
Figure 33: The Customize Feature calculates the mean from the mean values of the neighbors.

3.4.2 Execute the interpolation

Action!

1. Expand the Process sequence ‘**loop: while No. of unclassified** > **0**’.
2. Select the Process ‘**unclassified at _temp Level: ov_Feature Value = Mean_ov_Feature Value_2_objects with value**’ and **execute** it.
3. Go to the ‘Feature View’ window and browse to ‘**Object Features>Variables**’
4. Double-click on ‘**ov_Feature Value**’.

Rule Set Check



```
Process Tree
⊟ ▪ Working with LiDAR
   ⊟ ▪ Convert and correct
      ⊞ ▪ Convert LiDAR to new image layers
      ⊟ ▪ Interpolating LiDAR Number of Returns
            chess board: 1 creating '_temp Level'
            with Mean Number of Returns > 0  at  _temp Level: _object with value
            _object with value at  _temp Level: ov_Feature Value = Mean Number of Returns
         ⊟ ▪ loop: while No. of unclassified > 0
               unclassified at  _temp Level: ov_Feature Value = Mean_ov_Feature Value_2_objects  with value
               unclassified with ov_Feature Value > 0  at  _temp Level: _object with value
            at  _temp Level: Create temp. image layer 'Number of Returns Interpolated' using 'ov_Feature Valu
            at  _temp Level: delete
      ⊞ ▪ Interpolating LiDAR Elevation First
Main
```

Figure 34: Process Tree with Processes highlighted to interpolate.

Result Check

Objects with formally 0 values are updated and contain now the mean of the mean values of its neighbors.

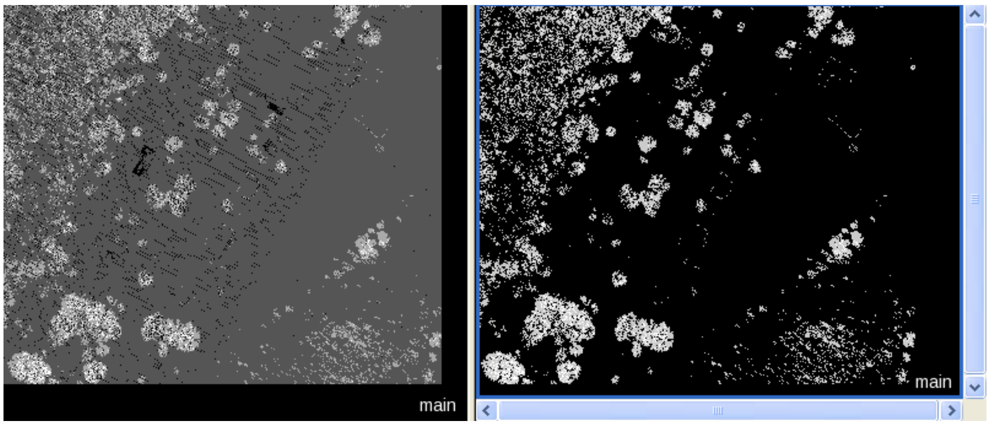


Figure 35: Left: Mean of ‘Number of Returns’; Right: updated Object variable.

Action!

5. Select and execute the Process ‘**unclassified with ov_Feature Value > 0 at _temp Level: _object with value**’.
6. Select and execute the Parent Process ‘**loop: while No. of unclassified > 0’**.

Result Check

All objects have now interpolated values for ‘Number of Returns’.

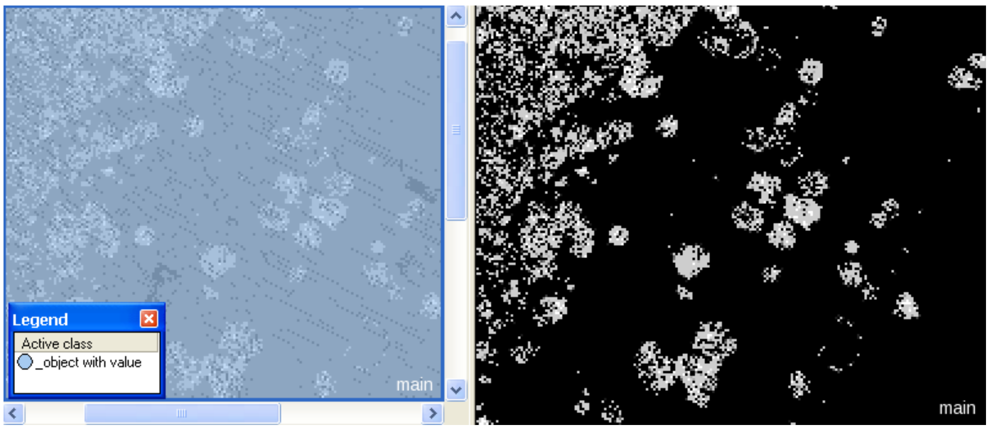


Figure 36: Left: Classification view'; Right: updated Object variable after executing the loop.

3.5 Creating the new image layer from Object Variable

This Chapter has the following sub-chapters

→ *The Process settings*

→ *Execute the Process and review the created image layer*

→ *Deleting the Object Level*

Information

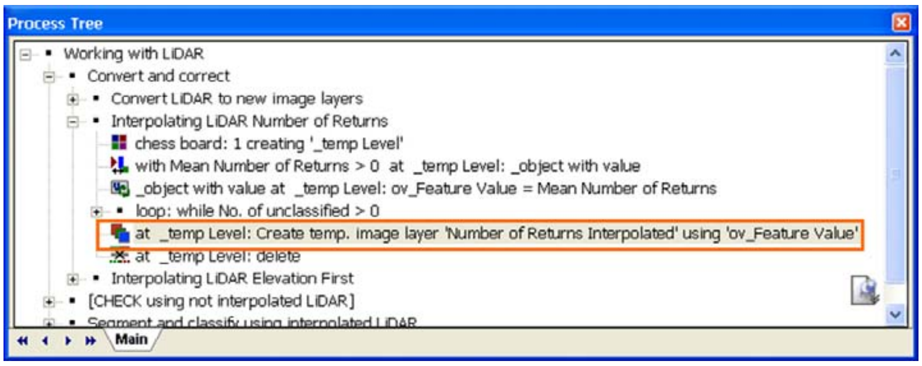
A new image layer 'Number of Returns Interpolated' is created from the values of the Object Variable 'ov_Feature Value' using the algorithm '**create temporary layer'**. This image layer will be then used for segmentation and classification instead of the original 'Number of Returns'.

3.5.1 The Process settings

Action!

Rule Set Check

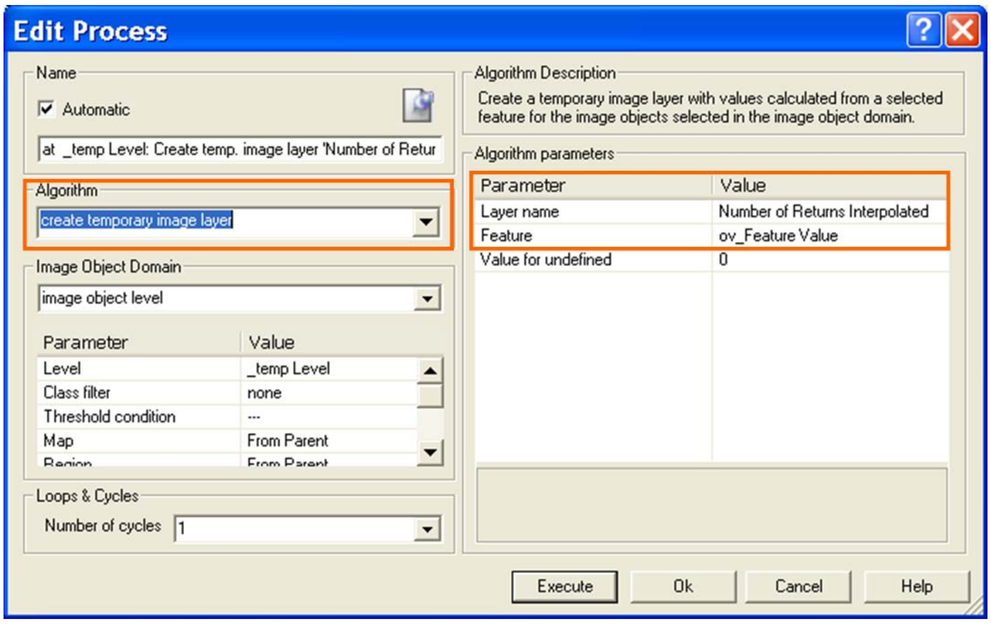
1. Collapse the Process sequence '**loop: while No. of unclassified > 0**'.
2. Double-click on the Process '**_object with value at _temp Level: ov_Feature Value = Mean Number of Returns**'.



```
Process Tree
- Working with LiDAR
  - Convert and correct
    + Convert LiDAR to new image layers
    - Interpolating LiDAR Number of Returns
        chess board: 1 creating '_temp Level'
        with Mean Number of Returns > 0 at _temp Level: _object with value
        _object with value at _temp Level: ov_Feature Value = Mean Number of Returns
      + loop: while No. of unclassified > 0
        at _temp Level: Create temp. image layer 'Number of Returns Interpolated' using 'ov_Feature Value'
        at _temp Level: delete
    + Interpolating LiDAR Elevation First
  + [CHECK using not interpolated LiDAR]
  + Segment and classify using interpolated LiDAR
Main
```

Figure 37: Process Tree with Process highlighted to create a temporary layer.

Settings Check



Name
☑ Automatic
at _temp Level: Create temp. image layer 'Number of Retur

Algorithm
create temporary image layer

Image Object Domain
image object level

Parameter	Value
Level	_temp Level
Class filter	none
Threshold condition	---
Map	From Parent
Region	From Parent

Loops & Cycles
Number of cycles 1

Algorithm Description
Create a temporary image layer with values calculated from a selected feature for the image objects selected in the image object domain.

Algorithm parameters

Parameter	Value
Layer name	Number of Returns Interpolated
Feature	ov_Feature Value
Value for undefined	0

Figure 38: Process settings to create the temporary layer 'Number of Returns Interpolated'.

The algorithm

- As algorithm 'create temporary image layer' is chosen. With this algorithm you can create an image layer containing feature values. It can be found in the 'Image layer operation' section of the algorithms list.

The Image Object Domain

- The Domain points to all objects of '**temp_Level**'

The Algorithm Parameters

- In the field 'Layer name' the name '**Number of Return Interpolated**' is defined.
- In the field 'Feature' the '**ov_Feature Value**' is defined.

3.5.2 Execute the Process and review the created image layer

Action!

3. Execute the Process '**_object with value at _temp Level: ov_Feature Value = Mean Number of Returns**'.

Result Check

The Object Variable is updated with interpolated values.

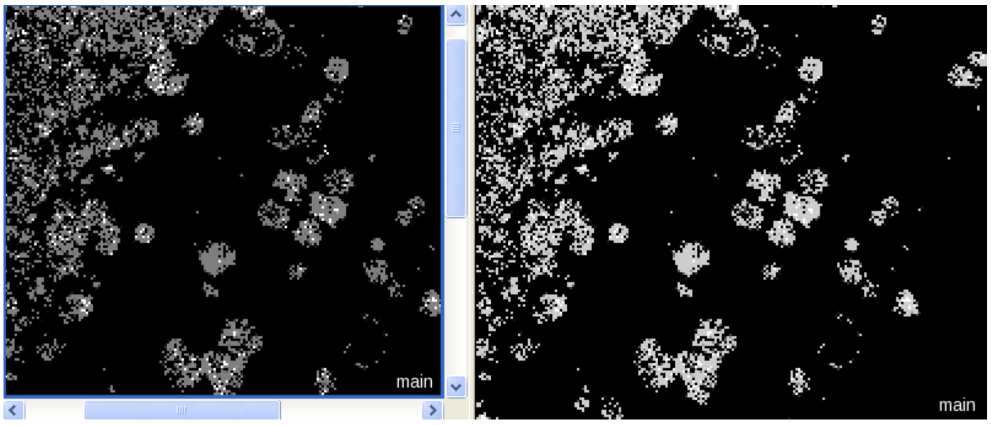


Figure 39: Left: Feature View for 'ov_Feature Value; Right: New created layer.

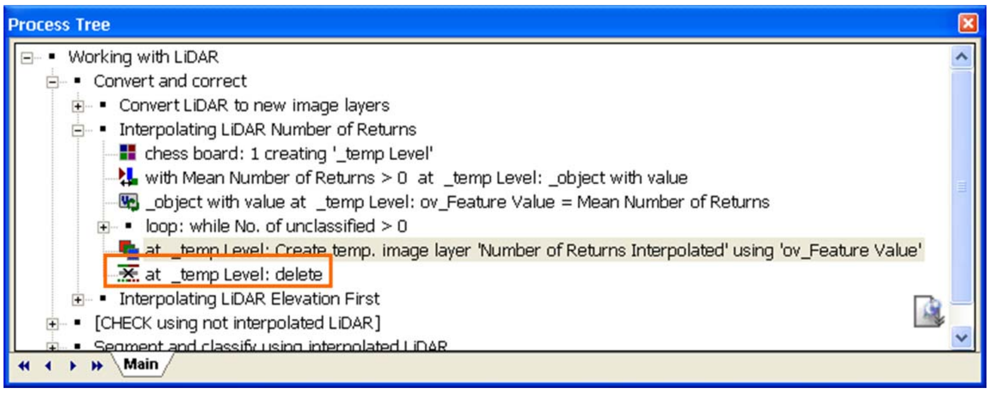
3.5.3 Deleting the Object Level

To be prepared for the next interpolation the existing Image Object Level is deleted.

Information

4. Execute the Process '**at _temp Level: delete**'.

Action!



Process Tree

```
⊟ · ▪ Working with LiDAR
    ⊟ · ▪ Convert and correct
        ⊞ · ▪ Convert LiDAR to new image layers
        ⊟ · ▪ Interpolating LiDAR Number of Returns
              chess board: 1 creating '_temp Level'
              with Mean Number of Returns > 0  at  _temp Level: _object with value
              _object with value at  _temp Level: ov_Feature Value = Mean Number of Returns
            ⊞ · ▪ loop: while No. of unclassified > 0
              at  _temp Level: Create temp. image layer 'Number of Returns Interpolated' using 'ov_Feature Value'
              at  _temp Level: delete
        ⊞ · ▪ Interpolating LiDAR Elevation First
    ⊞ · ▪ [CHECK using not interpolated LiDAR]
    ⊞ · ▪ Segment and classify using interpolated LiDAR
```

Figure 40: Process Tree with Process highlighted to delete the Image Object Level.

Rule Set Check

3.6 Creating the second interpolated layer for 'Elevation First'

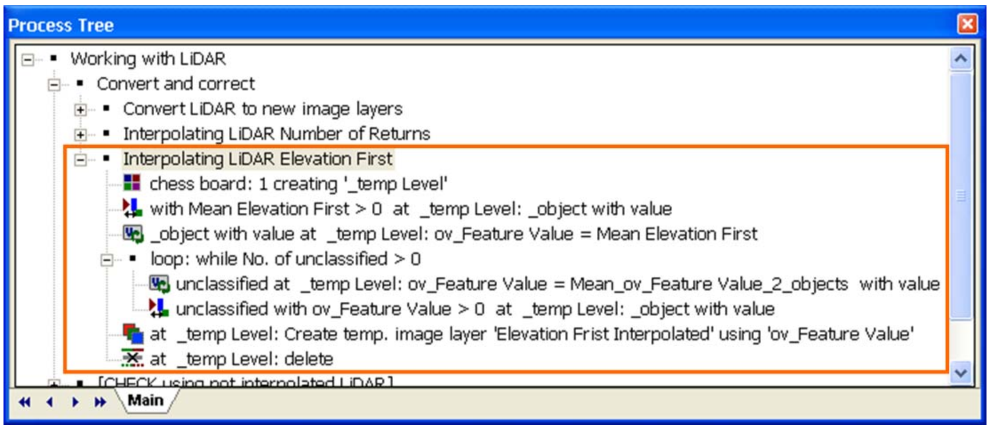
The same problems with 0 values also exist for the 'Elevation First' Layer. A similar interpolation Process sequence is applied to create an 'Elevation First Interpolated' layer.

Information

1. Collapse the Process sequence '**Interpolating LiDAR Number of Returns**'.
2. Expand the Process sequence '**Interpolating LiDAR Elevation First'**.
3. Execute the Parent Process '**Interpolating LiDAR Elevation First'**.

Action!

Rule Set Check



```
Working with LiDAR
  Convert and correct
    Convert LiDAR to new image layers
    Interpolating LiDAR Number of Returns
    Interpolating LiDAR Elevation First
      chess board: 1 creating '_temp Level'
      with Mean Elevation First > 0  at  _temp Level: _object with value
      _object with value at  _temp Level: ov_Feature Value = Mean Elevation First
      loop: while No. of unclassified > 0
        unclassified at  _temp Level: ov_Feature Value = Mean_ov_Feature Value_2_objects  with value
        unclassified with ov_Feature Value > 0  at  _temp Level: _object with value
      at  _temp Level: Create temp. image layer 'Elevation Frist Interpolated' using 'ov_Feature Value'
      at  _temp Level: delete
  [CHECK using not interpolated LiDAR]
```

Figure 41: Process Tree with Processes highlighted to interpolate the 'Elevation First'.

Result Check

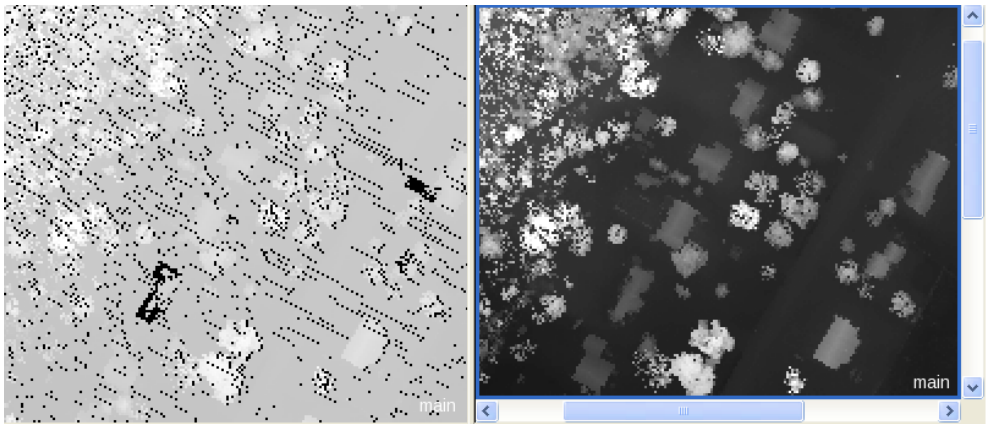


Figure 42: Left: Original, converted image layer; Right: new created, interpolated layer.

Lesson 4 Segmenting and classifying with the interpolated image layers

This Lesson has the following chapters

- → *Segment with interpolated image layers*
- → *Classify trees with interpolated image layers*
- → *Classify buildings with interpolated image layers*

4.1 Segment with interpolated image layers

All optical layers are used for segmentation. The two interpolated LiDAR images are used too, the weighting is set to 10 to balance the lower value range of these layers compared to the optical values.

As the layers are interpolated now, no pixel-sized Objects around the 0 values are created, the segmentation produces meaningful Objects.

Information

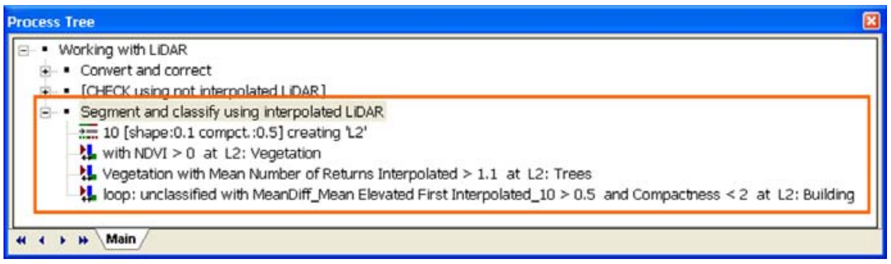
4.1.1 Preparation

1. In the Process Tree **collapse** the Parent Processes 'Convert and correct'.
2. **Expand** the Parent Processes '**Segment and classify using interpolated LiDAR**'.

Action!

✓

Rule Set Check



```
Process Tree
⊟ ▪ Working with LiDAR
  ⊞ ▪ Convert and correct
  ⊞ ▪ [CHECK using not interpolated LiDAR]
  ⊟ ▪ Segment and classify using interpolated LiDAR
       10 [shape:0.1 compct.:0.5] creating 'L2'
       with NDVI > 0  at  L2: Vegetation
       Vegetation with Mean Number of Returns Interpolated > 1.1  at  L2: Trees
       loop: unclassified with MeanDiff_Mean Elevated First Interpolated_10 > 0.5  and Compactness < 2  at  L2: Building
Main
```

Figure 43: Process Tree with Processes highlighted to segment and classify using the interpolated data.

4.1.2 Process settings

Settings Check

- Both **interpolated LiDAR** layers are used for segmentation and weighted with 10.

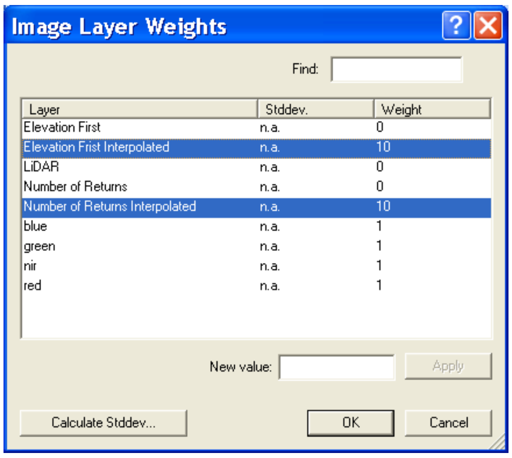


Image Layer Weights

Find:

Layer	Stddev.	Weight
Elevation First	n.a.	0
Elevation Frist Interpolated	n.a.	10
LiDAR	n.a.	0
Number of Returns	n.a.	0
Number of Returns Interpolated	n.a.	10
blue	n.a.	1
green	n.a.	1
nir	n.a.	1
red	n.a.	1

Figure 44: ‘Image Layer Weights’ dialog box with both interpolated image layers selected.

4.1.3 Execute and review the Result

Action!

1. **Execute** the Process ‘10 [shape:0.1 compct.:0.5] creating 'L2"
2. Switch on the outlines and review the created Objects.
3. Display in one viewer the segmentation of the main Map, in the other the segmentation of the ‘Map Temp’.

Result Check

The Objects are created in a meaningful way.

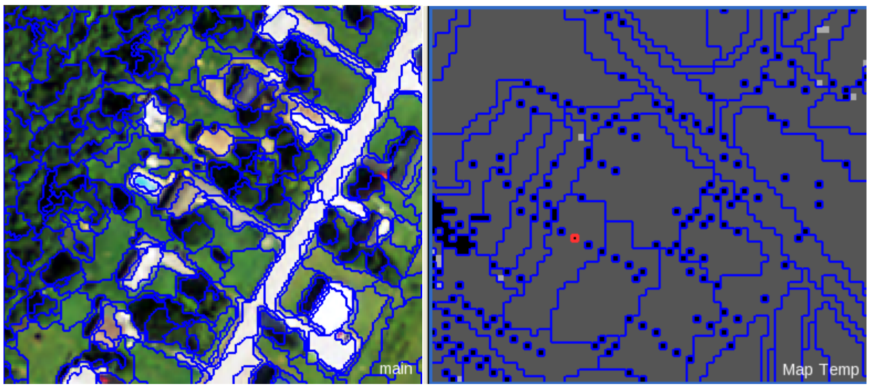


Figure 45: Left: Objects created using the interpolated image layer (main Map); Right: Objects created using the not interpolated image layer (Map Temp).

4.2 Classify trees with interpolated image layers

Information

The mean value of the image layer ‘Number of Returns Interpolated’ is used to classify Trees. As the 0 values are corrected no holes are in the classification.

Action!

1. **Execute** the Process ‘with NDVI > 0 at L2: Vegetation’ and ‘**Vegetation with Mean Number of Returns Interpolated > 1.1 at L2: Trees**’.

2. Switch on the classification view and review the classification.

The trees are classified without holes

Result Check

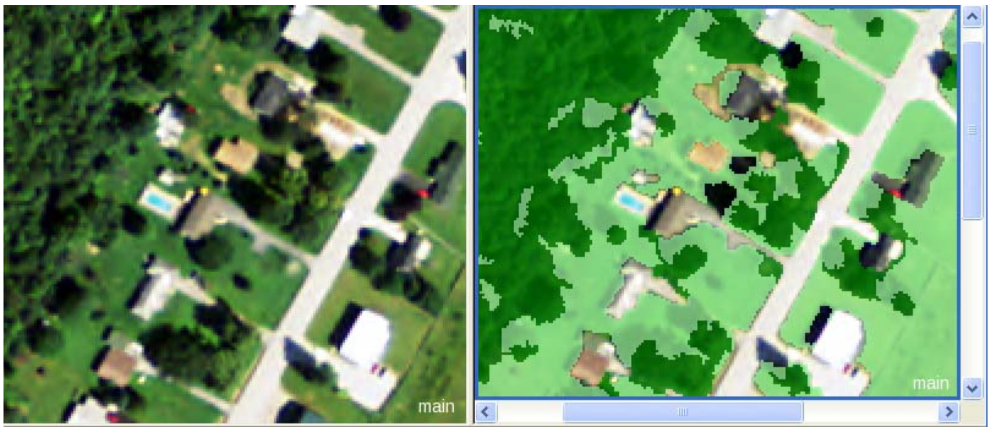


Figure 46: Left: Optical layers; Right: Classification Result using **interpolated** image layer 'Number of Returns Interpolated' (mainMap).

Compare to result using not interpolated image layer:

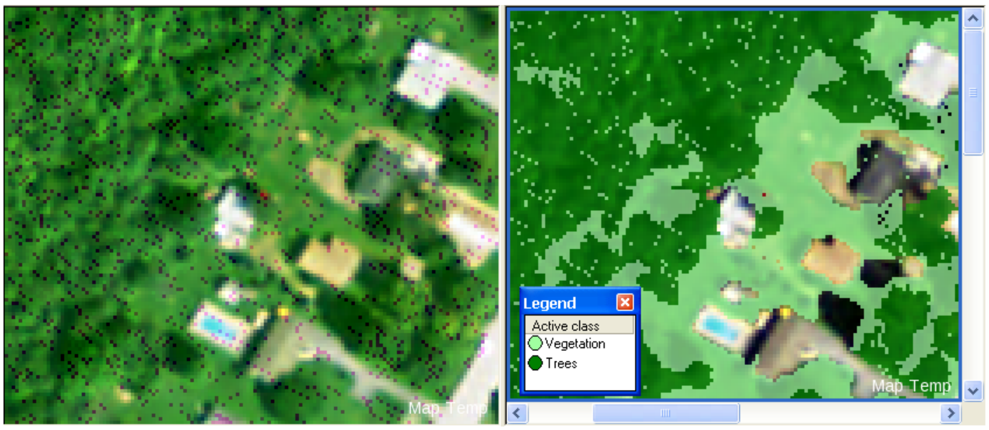


Figure 47Left: Optical layers and Number of Returns **not interpolated** in mixed view; Right: Classification Result (Map Temp).

4.3 Classify buildings with interpolated image layers

Information

Objects with difference in elevation in comparison to their neighbors are classified as Building.

As the 0 values of the not interpolated image layers are corrected, the Mean difference calculation gives back correct values fitting for Building objects, respectively other objects actually not being a building do not fulfill the condition. No misclassifications are the result.

Action!

1. **Execute** the Process 'loop: unclassified with MeanDiff_Mean Elevated First Interpolated_10 > 0.5 and Compactness < 2 at L2: Building'.
2. Switch on the classification view and review the classification.

Buildings are classified correctly, no misclassification of e.g. the road.

Result Check

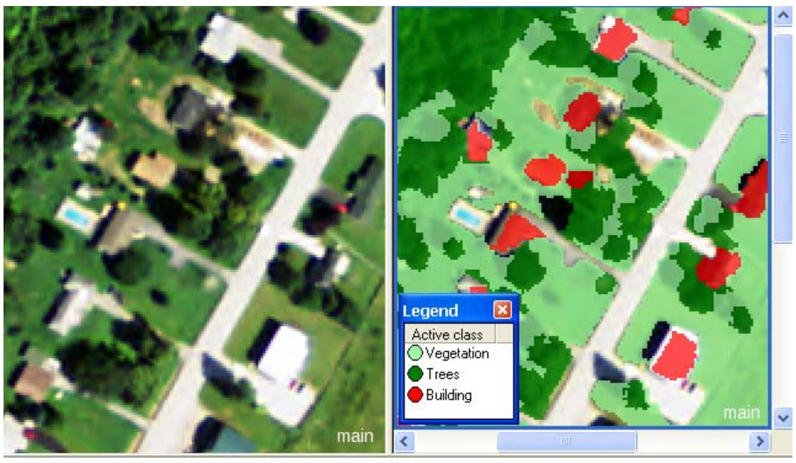


Figure 48: Left: Optical layers; Right: Classification Result using interpolated image Layers (main Map).

Compare to result using not interpolated image layer:

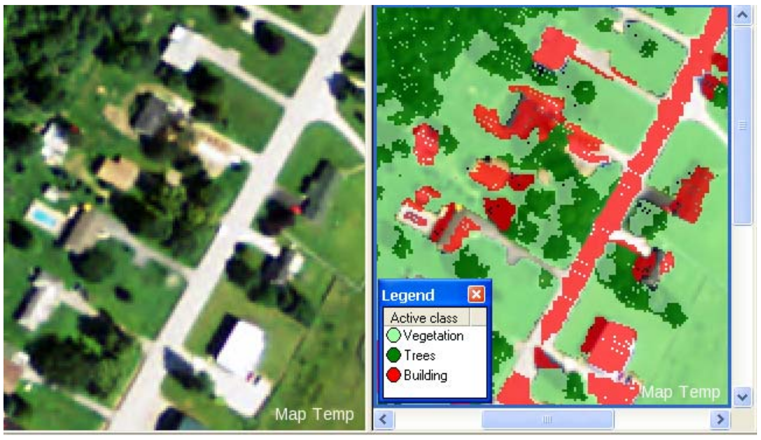


Figure 49: Left: Optical layers; Right: Classification Result using not interpolated image Layers (Map Temp'.