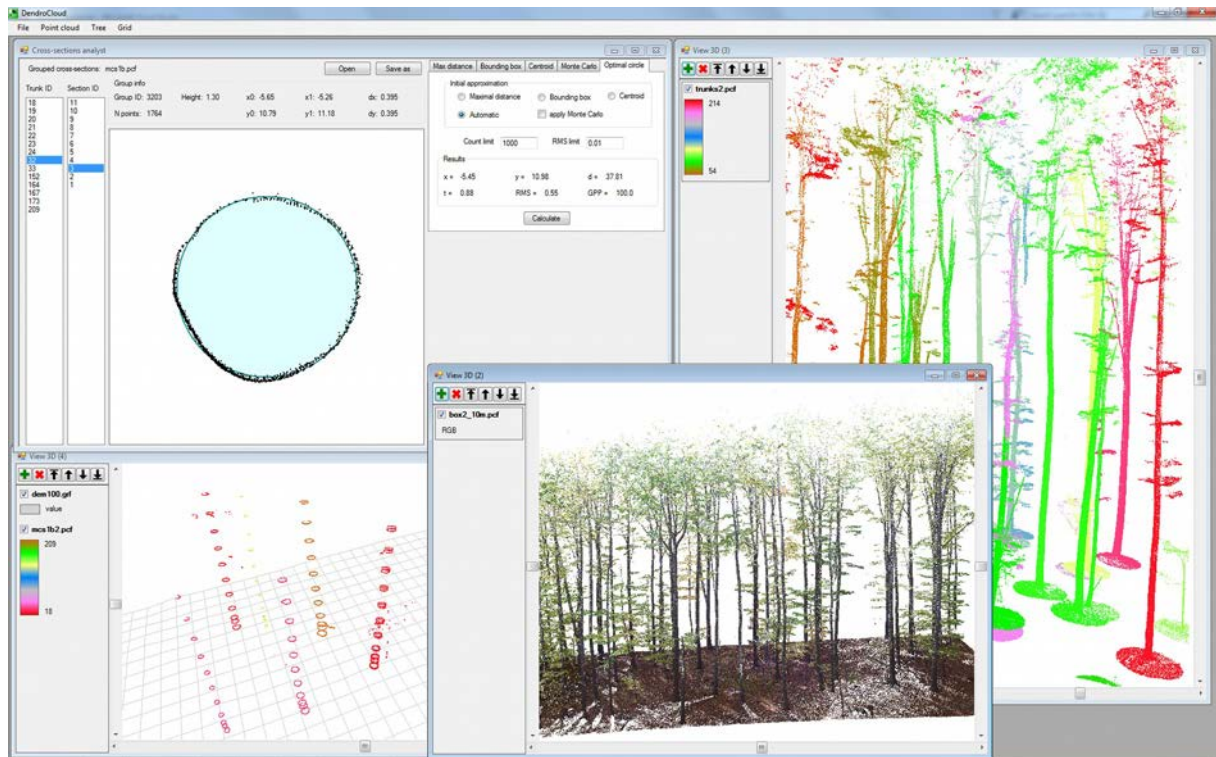




DendroCloud

point cloud processing software for forestry

version 1.45



Milan Koreň

Technical University in Zvolen

gis.tuzvo.sk/dendrocloud/

Contents

| | |
|--|----|
| 1 DendroCloud..... | 4 |
| 1.1 Introduction..... | 4 |
| 1.2 End user licence agreement..... | 4 |
| 1.3 Download and installation..... | 5 |
| 1.4 System configuration..... | 5 |
| 2 Point cloud import and export..... | 6 |
| 2.1 Import XYZ..... | 6 |
| 2.2 Import CSV..... | 7 |
| 2.3 Import TXT..... | 8 |
| 2.4 Import LAS..... | 9 |
| 2.5 LAS information..... | 10 |
| 2.6 Export CSV..... | 11 |
| 2.7 Export VRML..... | 12 |
| 2.8 XY reduction..... | 13 |
| 3 Multi-temporal three-dimensional grid..... | 14 |
| 3.1 Create new grid..... | 15 |
| 3.2 2D grid from point cloud..... | 16 |
| 3.3 3D grid from point cloud..... | 17 |
| 3.4 Import grid from text file..... | 18 |
| 3.5 Export grid to text file..... | 19 |
| 3.6 Grid overlay..... | 20 |
| 4 Point cloud processing..... | 21 |
| 4.1 Filters..... | 21 |
| 4.1.1 Filter by box..... | 21 |
| 4.1.2 Filter by cylinder..... | 22 |
| 4.1.3 Trim..... | 23 |
| 4.1.4 Cross-section..... | 24 |
| 4.1.5 Count in cones..... | 26 |
| 4.1.6 Save by ID..... | 27 |
| 4.2 Digital terrain model..... | 28 |
| 4.2.1 Vertical projection..... | 28 |
| 4.2.2 Extract surface..... | 29 |

| | |
|---|----|
| 4.3 Merge point clouds..... | 30 |
| 4.4 Resample point cloud..... | 31 |
| 4.5 Convert to 3D grid..... | 32 |
| 4.6 Group by distance..... | 33 |
| 5 Tree identification and measurement..... | 34 |
| 5.1 Cross-sections analyst..... | 34 |
| 5.2 Tree diameter estimation..... | 37 |
| 5.2.1 Minimum bounding box method..... | 38 |
| 5.2.2 Centroid method..... | 39 |
| 5.2.3 Maximum distance method..... | 40 |
| 5.2.4 Monte Carlo method..... | 41 |
| 5.2.5 Optimal circle method..... | 42 |
| 5.3 Extract trunks..... | 43 |
| 5.4 Tree base estimation..... | 44 |
| 5.5 Multiple cross-sections..... | 47 |
| 6 Visualization..... | 50 |
| 6.1 3D View..... | 50 |
| 6.2 Point cloud theme..... | 52 |
| 6.3 2D grid theme..... | 54 |
| List of abbreviations..... | 56 |
| Appendix A: Data types..... | 57 |
| Appendix B: Extended ASCII raster format..... | 58 |
| Appendix C: Standard fields of a point cloud..... | 60 |

1 DendroCloud

1.1 Introduction

Software **DendroCloud** is designed to process point cloud data from terrestrial laser scanning (TLS) or close-range photogrammetry for forestry purposes.

Main functionality of the software:

- ◆ **Point cloud processing:** imports, filters, spatial clustering, horizontal cross-sections.
- ◆ **Digital elevation models:** grid creation, import and export, vertical projections, points extraction, grid operations.
- ◆ **Three-dimensional visualization:** point cloud, digital elevation model, grid.

We appreciate any comments and suggestion to its improvement. If you have any, please send them by e-mail to the author.

1.2 End user licence agreement

The licence

Permission is hereby granted to any person obtaining a copy of this software and associated documentation files to use the software and documentation (the "software") free of charge.

The software is provided "as is" and the author disclaims all warranties with regard to this software including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, direct, indirect, or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

The disclaimer

The software is provided "as is", without warranty of any kind.

You use the software at your own risk.

The software is a work in progress and may contain some defects. We do not guarantee that the software will be error-free, the results obtained from the use of the software will be accurate and reliable.

We will not be liable for any direct or indirect damages resulting from the use or misuse of the software, even if the author has been advised of the possibility of such damage.

The agreement

By downloading or using the software, you agree to be bound by the terms and conditions of this end user licence agreement. If you do not agree to the terms and conditions of this end user licence agreement, you may not download or use the software.

1.3 Download and installation

Read carefully the end user licence agreement.

Download the latest version of the software from its home page gis.tuzvo.sk/dendrocloud/.

Extract all files from the archive to the output directory.

Run the program **DendroCloud.exe**.

1.4 System configuration

Program configuration parameters are stored in file **dendrocloud.cnf**. Configuration file is usually stored in the program's directory. Configuration file contains parameters, which are needed to set usage of processors, memory, graphics card, etc.

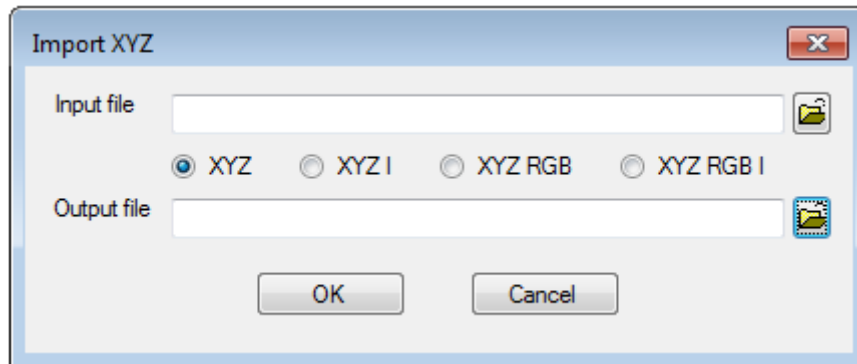
| | |
|--------------------------------|--|
| DendroCloud.Licence:="" | A licence key of the application. |
| DendroCloud.MaxProcesses:=-1 | Maximum number of threads for a parallel processing. Default value -1 means, that all available cores can be used. |
| Grid.SubgridMethod:=1 | Code of grid subdivision method for a parallel processing. |
| Grid.SubgridFactor:=3 | Number of tasks for a core. |
| Grid.SubgridMinColumns:=10 | Minimum number of columns for subdivision of an input grid. |
| Grid.SubgridMinRows:=10 | Minimum number of rows for subdivision of an input grid. |
| Grid.TraceDstBufferSize:=10000 | Grid cache size for distance analyses. |
| Pcld.ThemeMaxPointsM:=10 | Maximum number of points for visualization in millions. Input point cloud will be resampled to specified number of points. Parameter is set up in accordance to a graphics card performance. |
| Pcld.MaxBufferLength:=3000000 | Maximum size of point cloud cache. |
| Pcld.XYZDataType:="float" | Data type of point coordinates (float, or double) |

2 Point cloud import and export

Point clouds can be imported from text, CSV and LAS files.

2.1 Import XYZ

Text files are the most simple way to transfer data. They can be created in different programs. Module **Import XYZ** is used to import point clouds from a text file in defined formats.



The input text file must contain data in one of supported formats:

- x y z: point coordinates
- x y z i: point coordinates and intensity of reflected laser beam
- x y z r g b: point coordinates and colour
- x y z r g b i: point coordinates, colour and intensity of reflected laser beam

Point coordinates are stored as decimal numbers (*float*, or *double*).

Intensity of reflected beam is *short integer*.

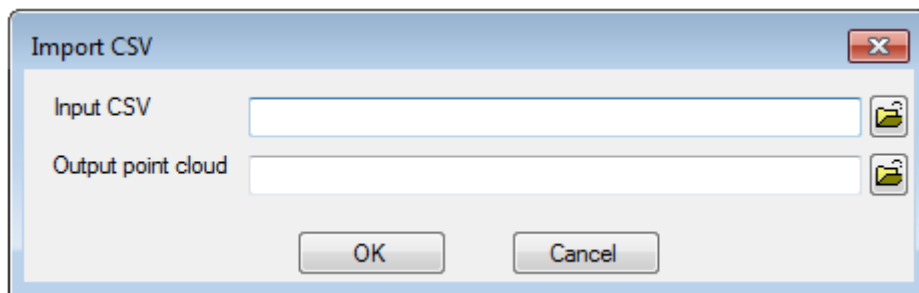
Colour components are *unsigned bytes*.

There is no header row in the input text file. Each row contains data just for one point. The order of fields cannot be changed. Numbers in a row must be separated by one character (space, semicolon, or tab). The input text file cannot contain comments nor empty lines.

2.2 Import CSV

The first row of a Comma Separated Value (CSV) file is called a header row. The header row contains names of fields. Field name is a string starting by a letter. Field name cannot contain separator character. Structure of a CSV file is variable, it can be defined by a user. Main disadvantage of CVS files is absence of fields' data types specification.

The fields x, y, z are required for a CSV file containing a point cloud. Additional point attributes (colour, intensity, height above ground) can be included. The order of fields is not fixed. The import procedure automatically recognizes standard fields (x, y, z, r, g, b, i) of a point cloud and assigns them to appropriate data types. The other fields are stored as decimal numbers (*double*).



Files in CSV format are imported by the module **Import CSV**. The user enters a name of the input CSV file name and a name of the output point cloud. The names of output point cloud fields correspond to the field names of the input CSV file.

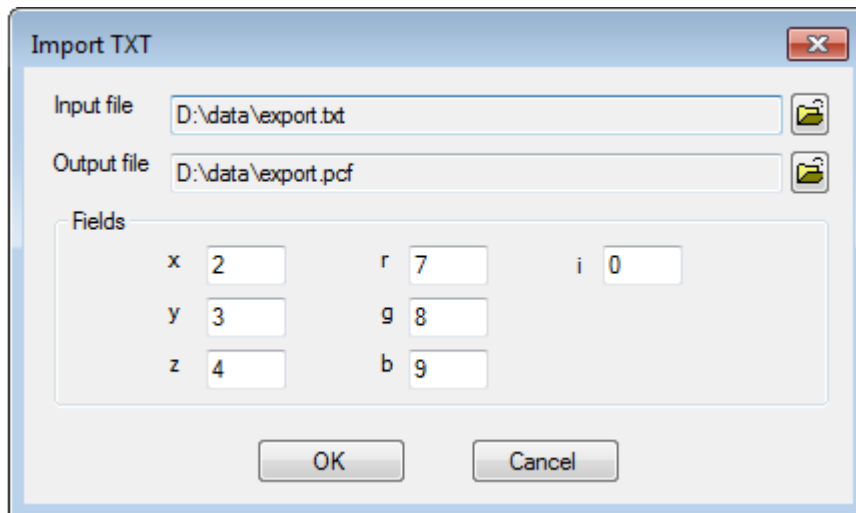
2.3 Import TXT

Some point cloud processing programs export point clouds with additional fields, or fields are arranged in a different order than XYZ files. If the input text file does not contain a header row, fields cannot be automatically recognized.

The module **Import TXT** reads text files without header rows with user-defined structure. The structure of the input file is specified in the form. A column's index is assigned to each of the standard point cloud fields. Columns are numbered from 1. Index of a missing column is 0.

Example

Your input text file contains fields *id*, *x*, *y*, *z*, *angle*, *distance*, *r*, *g*, *b*. The columns *id*, *angle*, *distance* must be skipped. There is no field of reflected intensity (*i*). The input file can be imported by the following settings:

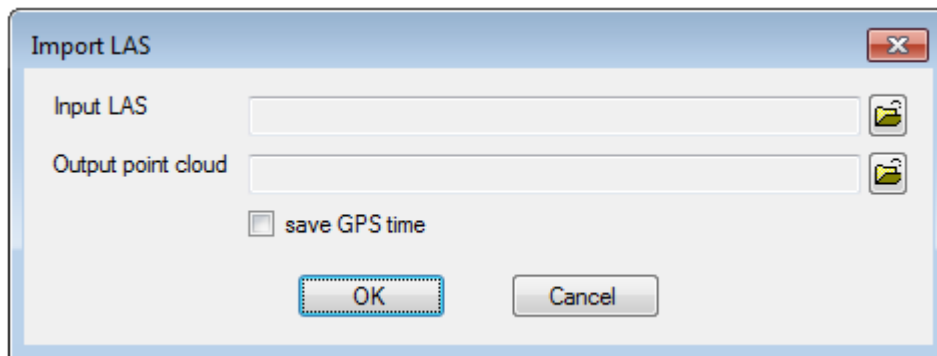


| Field | Index |
|-------|-------|
| x | 2 |
| y | 3 |
| z | 4 |
| r | 7 |
| g | 8 |
| b | 9 |
| i | 0 |

2.4 Import LAS

The LAS is an open file format for exchange of point cloud data developed and maintained by ASPRS (asprs.org). It is a binary file format designed for specific needs of LIDAR data interchange. The LAS format is widely supported by vendors of software for processing aerial and terrestrial laser scanning data.

The latest version of LAS format (1.4, July 2013) supports 10 point data record formats. Projection information, waveform packet information, user-defined data, and other metadata can be also stored in LAS variable length records.

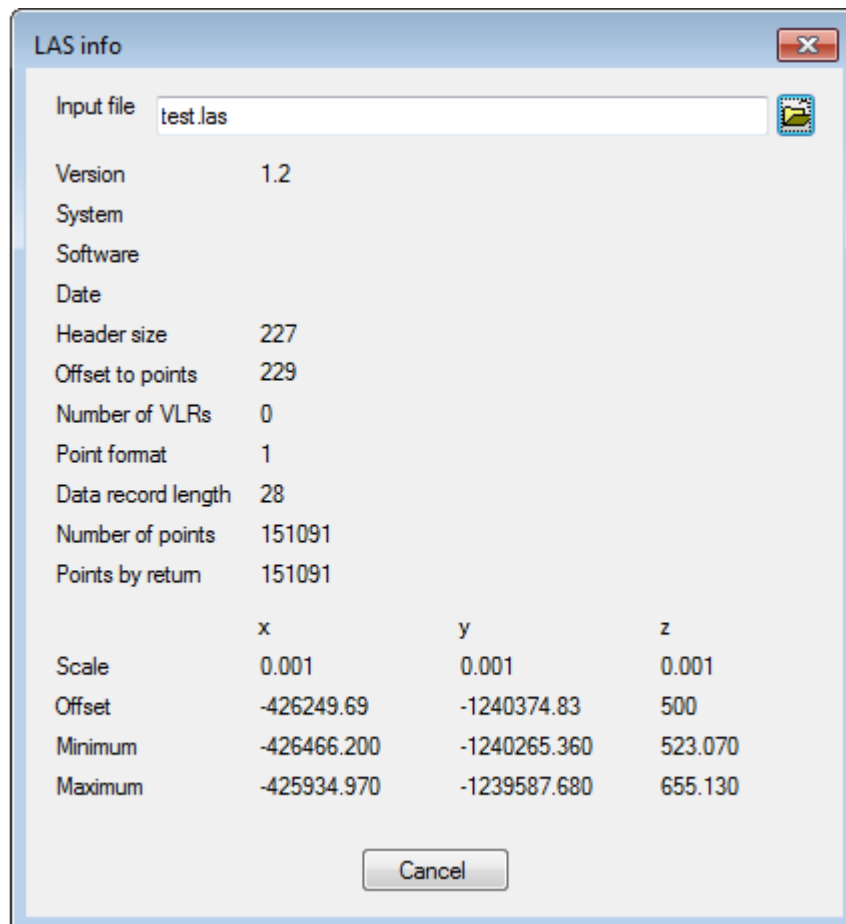


The module **Import LAS** converts point data from a LAS file to point cloud file (*.pcf, *pcd). The output point cloud file consists of six main fields and one optional field:

- x, y, z: point coordinates (*float* or *double*, specified by *Pcld.XYZDataType* parameter in the configuration file),
- i: intensity (*int*),
- rn: return number (*int*),
- nor: number of returns (*int*),
- gps_time: optional field, used in aerial and mobile laser scanning (*double*).

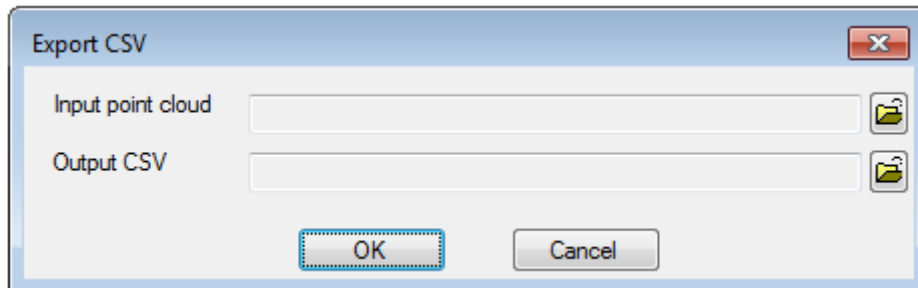
2.5 LAS information

The utility **LAS Info** displays detailed information from a LAS file. The module reads the header of a given LAS file and prints out information about file format version, system identifier, generating software, date of creation, header size, number of variable length records (VLR), point record format, point record length, number of point records, number of points by return, transformation parameters (scale, offset), and extents of the point cloud coordinates.



2.6 Export CSV

The utility **Export CSV** saves all point cloud fields to a text file. The first row of the output text file contains the names of fields. Attributes of each point are saved as a separate line. The CSV file can be easily imported into spreadsheet calculator, statistical programs or the other point cloud processing software.

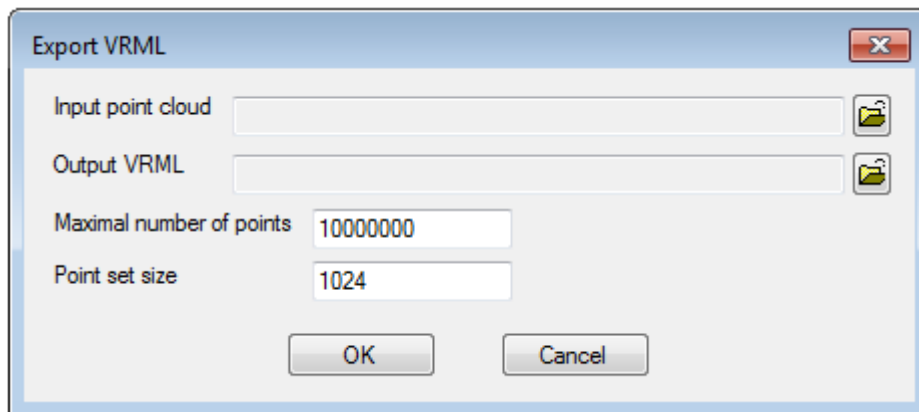


The example of the exported point cloud data:

```
x y z r g b i group_id trunk_id sec_id sec_h
6.0529 -11.3576 526.6115 57 50 44 1165 11 1 1 1.3000
8.2279 21.3381 532.5719 56 46 45 1327 21 2 1 1.3000
-5.4194 23.5161 531.5906 53 40 43 1386 31 3 1 1.3000
8.2291 21.3387 532.5627 54 46 44 1314 21 2 1 1.3000
-5.4594 23.3991 531.4923 24 18 6 1362 31 3 1 1.3000
5.8614 21.3588 532.3185 57 48 51 1221 41 4 1 1.3000
5.9221 21.4093 532.3155 54 43 43 1396 41 4 1 1.3000
3.0882 -9.2061 526.8158 53 38 41 1410 51 5 1 1.3000
9.8408 -5.7722 527.7593 50 39 42 1379 61 6 1 1.3000
-20.7818 -32.4589 519.8773 34 28 28 1453 71 7 1 1.3000
```

2.7 Export VRML

Virtual Reality Modelling Language (VRML, www.w3.org/MarkUp/VRML/) is a public text file format used to represent 3D scenes. A virtual world is represented by a set of graphical objects, like 3D surfaces, textures, polygons, vertices. Properties and events can be associated with graphical components of a 3D scene. A VRML file can be interactively viewed in a VRML viewer, browser plug-in (openvrml.org) or virtual reality (etools.tuzvo.sk/cave/index_en.html).



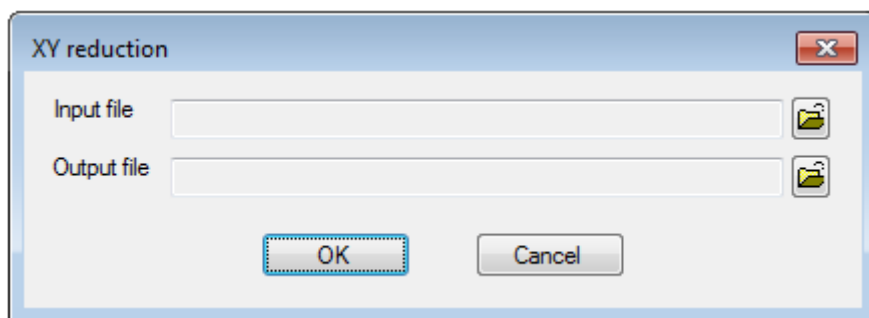
The module **Export VRML** generates a VRML file from a given point cloud. The input point cloud is resampled to a specified number of points. Points are stored in point sets of specified sized. Coordinates and colour of each point are saved to the output VRML file.

The example of the generated VRML file with 5 points:

```
#VRML V2.0 utf8
Shape { geometry PointSet {
  coord Coordinate { point [
    29.9612 532.7857 -13.5027,
    29.9606 532.7859 -13.5063,
    29.9594 532.7861 -13.5107,
    29.9584 532.7863 -13.5148,
    29.9573 532.7866 -13.5192
  ] }
  color Color { color [
    0.4235 0.4392 0.6902,
    0.4196 0.4353 0.6863,
    0.4157 0.4392 0.6824,
    0.4196 0.4431 0.6863,
    0.4235 0.4471 0.6824
  ] }
}}
```

2.8 XY reduction

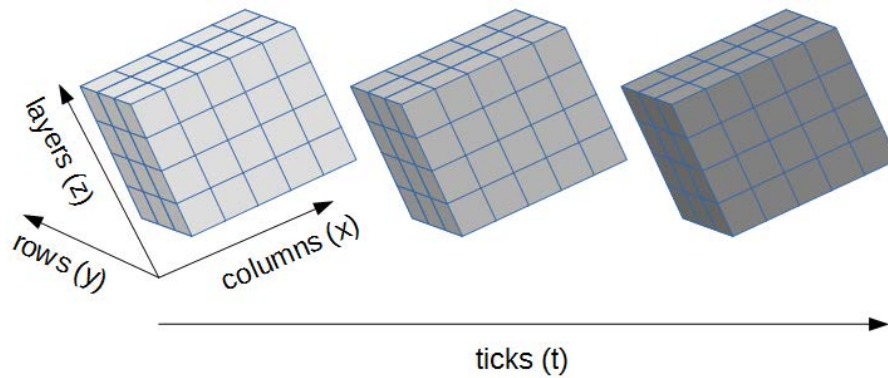
The **XY reduction** is a utility, which makes coordinates of points smaller. The algorithm just the point cloud to the origin of coordinate system without any distortions. Only x and y coordinates are shifted. The z-coordinate of points is not transformed. The coordinate transformation preserves the extent of a point cloud and relative positions of points. In the other words, the model is unchanged and represented by smaller coordinates.



A reduction of x and y coordinates is needed in cases, when coordinates are too large for correct visualisation. For example, point cloud from a mobile laser scanning is provided in UTM coordinate system. The extent of scanned area is usually several kilometres, but x-coordinates of points are greater than 5 millions. Point cloud coordinates are stored as float number with limited precision. In such situation the data type of point cloud coordinates must be set to double (parameter *Pcld.XYZDataType* in configuration file, see [1.3 System configuration](#)). The point cloud coordinates will be correctly imported and processed, but computer graphics also uses float numbers for coordinates of geometrical objects. The loss of coordinates precision results in fallacious display of the point cloud on a computer monitor. The reduction of x and y coordinates helps to improve quality of point cloud visualisation.

3 Multi-temporal three-dimensional grid

A point cloud can be converted to a 3D grid. Grids are used for digital terrain models (DTM) and digital surface models (DSM). DTM and DSM are important layers in the processing point clouds for forestry purposes. DTM is used during the processes of vegetation segmentation, cross-sections generation, tree identification. Tree height can be estimated by a canopy height model (CHM), which is calculated as a difference between DSM and DTM.



The GIS3DT library supports processing of multi-temporal three-dimensional grids. Grids are divided to columns along x-axis, rows along y-axis, and layers along z-axis. Ticks represent division of time axis.

Different types of numbers (*byte*, *integer*, *long integer*, *float*, *double*) can be stored in the grid. The library provides procedures for import, export, processing, analysis and visualisation of three-dimensional dynamic grids. Some of these functions and tools are included in the **DendroCloud** software.

3.1 Create new grid

A new empty 3D grid is created by the module **Create new grid**. The user enters name of the output grid, spatial framework, null value and initial value of the new grid. The spatial framework is given by a grid extent and number of cells in each dimension. The data type of the new grid is [double](#). The created grid is filled by the initial value. The initial value is *null* by default. The null value means, that the cell is empty (no numeric value is stored in a cell).

Create new grid

Output grid

☐ Min ☐ Max ☐ Cells ☒ Cell size

x 0 500 50 10

y 0 500 50 10

z 0 500 50 10

t -0.5 0.5 1 1

☒ Use default null value ☒ Initialize with null value

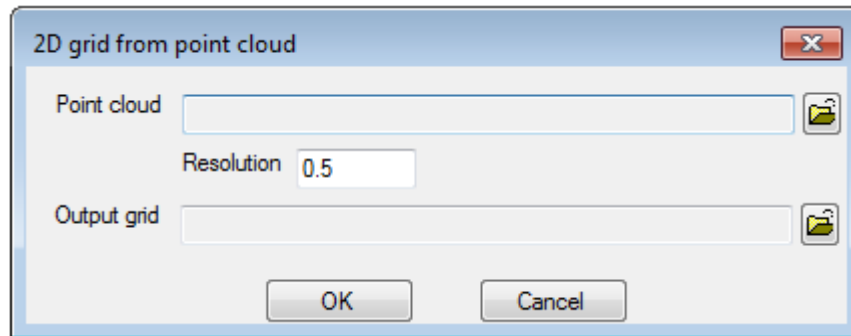
New null value -9999

Initial value 0

OK Cancel

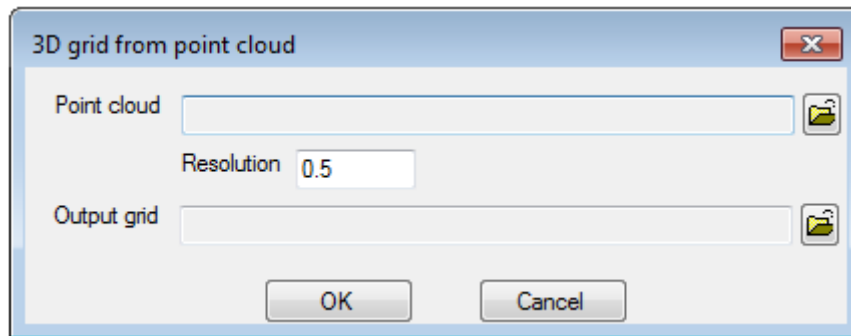
3.2 2D grid from point cloud

The utility **Create grid from point cloud** is used to create a new 2D grid, whose extent corresponds to a given point cloud. The width and the height of the output grid's cell are equal to a given resolution. The data type of the created grid is [float](#). The null value of the grid is set to default and the grid is initialized by the null value.



3.3 3D grid from point cloud

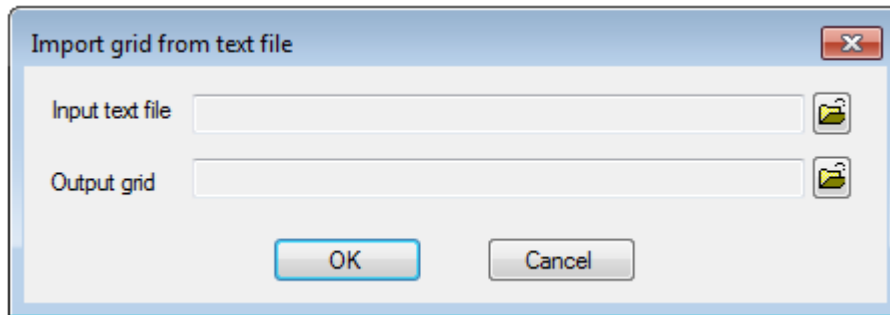
The utility **2D grid from point cloud** creates a new 3D grid, whose extent corresponds to a given point cloud. The size of the output grid's cell are equal to a given resolution. The data type of the created grid is [*float*](#). The null value of the grid is set to default and the grid is initialized by the null value.



3.4 Import grid from text file

The module **Import grid from txt** reads data from a text file in the [extended ASCII raster format](#). The procedure reads the header of the input text file, creates a new grid, and fill the output grid with data from the exchange text file. The data type of the created grid is [double](#).

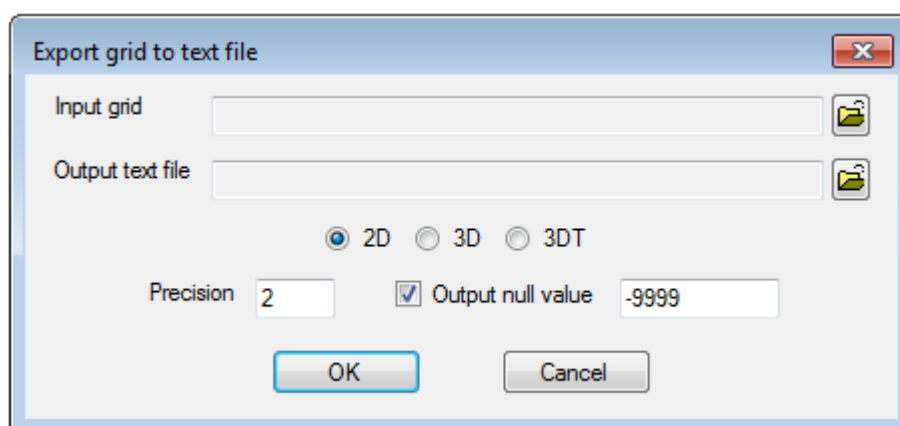
The procedure supports the [extended ASCII raster format](#), which enables to import and export three-dimensional multi-temporal grids. The number of dimensions of the input grid is recognized automatically.



3.5 Export grid to text file

The Module **Export grid to txt** writes a grid to the [extended ASCII raster format](#). The user specifies a name of the input grid, a name of the output text file, a format of the output file, a precision (number of decimal places) of the output numbers, and an output null value (the default null value is -9999).

The format of a two-dimensional raster (2D) is compatible with ESRI raster ASCII format (www.esri.com). Three-dimensional and multi-temporal grids can be exported too. The additional meta-information about higher dimensions is added into the output text file header.



If the input grid is three-dimensional (3D) and 2D format of the output file is selected, only the first layer of the grid is exported to the text file.

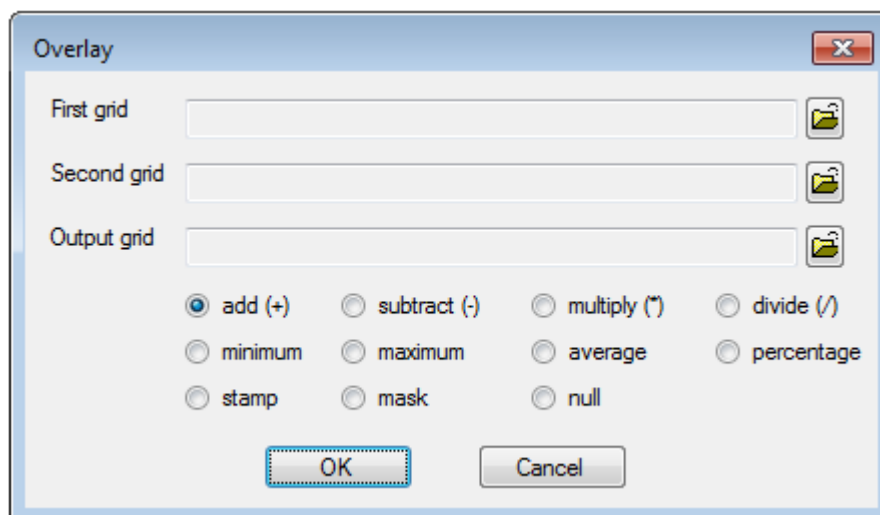
If the input grid is three-dimensional (3D) and 2D format of the output file is selected, only the first layer of the grid is exported to the text file.

If the input grid is three-dimensional multi-temporal (3DT) and 2D format of the output file is selected, only the first layer of the first tick is exported to the output file.

If the input grid is three-dimensional multi-temporal (3DT) and 3D format of the output file is selected, only the first tick is exported to the text file.

3.6 Grid overlay

Main operations with 3DT grids are implemented in the module **Overlay**. The input grid must have the same spatial reference (dimension, extent, number of ticks, layers, rows and columns). The selected operation is performed for each cell and the result is saved into the output grid.



Arithmetical operations:

- addition,
- subtraction,
- multiplication,
- division,
- minimum value,
- maximum value,
- average of values,
- percentage.

Overlay operations:

- *stamp*: all not null values of the first grid are transferred to the corresponding cells of the second grid. The values of the other cells of the second grid remain unchanged.
- *mask*: all null values from the first grid are transferred to the second grid. In other words, cell values in the second grid remain unchanged wherever the cells of the first grid are not null.
- *null*: cells of the second grid are set to null wherever the cell value of the first grid is not null. The other cells of the second grid are unchanged.

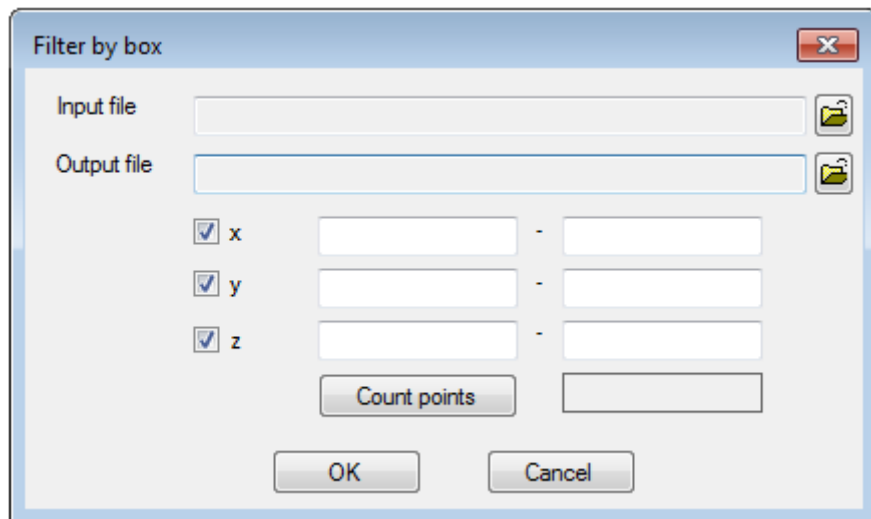
If an error occurs during the operation (e.g. division by zero), the null value will be written to corresponding cell of the output grid.

4 Point cloud processing

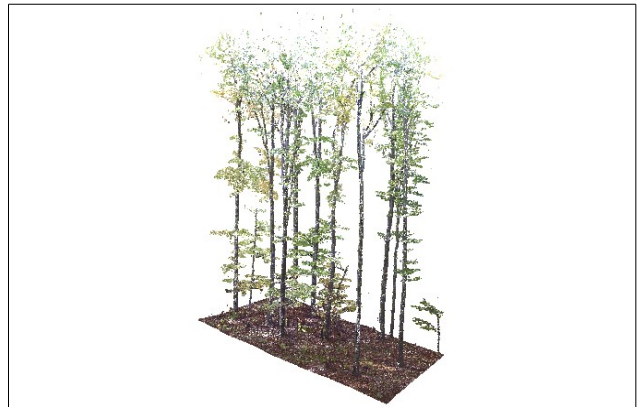
4.1 Filters

4.1.1 Filter by box

The Filter by box is a main and very often used method of a point cloud filtering. The algorithm selects point within a defined extent of coordinates. The points which fall inside the block are saved to an output point cloud. All points' attributes are copied to the output file.

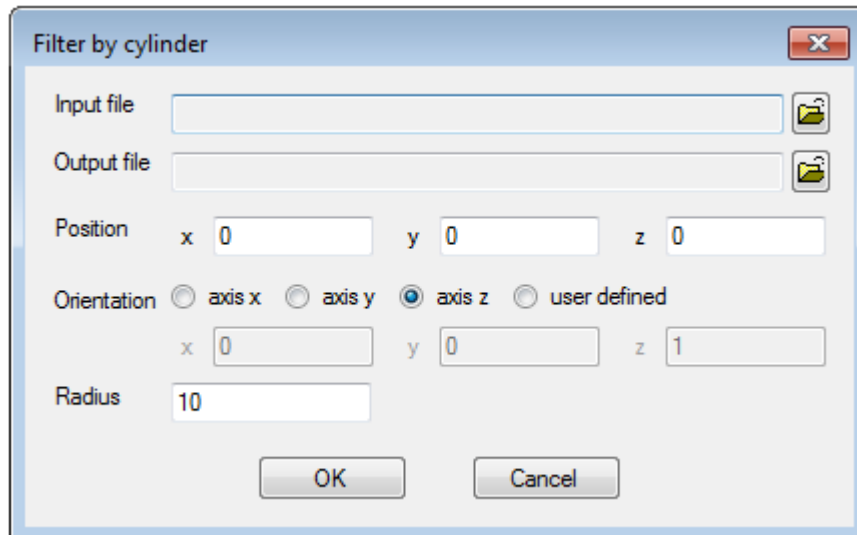


Examples:

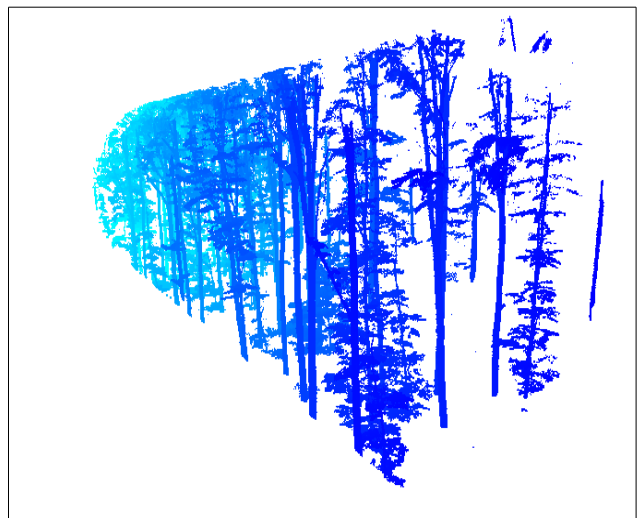


4.1.2 Filter by cylinder

The utility Filter by cylinder extracts all points from an input point cloud, which fall inside a given cylinder. The cylinder is defined by its position, axis orientation and radius.



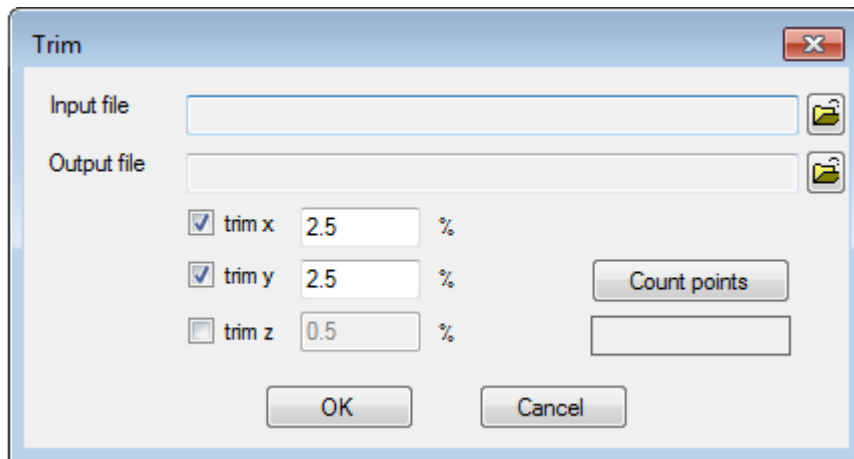
Examples:



4.1.3 Trim

The range of a modern terrestrial laser scanner is several tens or hundreds metres. In a forest environment a point is often recorded in distance greater than 100 metres from a scanner position. A density of the point cloud decreases rapidly with an increasing distance from a scanner.

The **Trim** module extracts points from the scanned area with the highest density of points. Distributions of points along main coordinate axes are calculated. A specified percentages of points are dropped from each side along axes. For example, if 2.5% is specified for x-axis, then a new extent in x-direction will contain 95% of points. Tails of point cloud, which are distant and have a low density of points are removed.



Example:



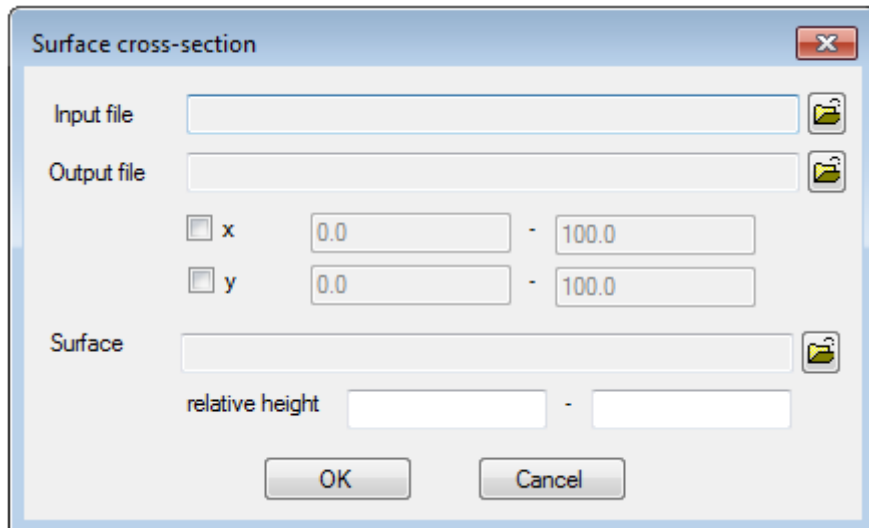
The original point cloud



The trimmed point cloud

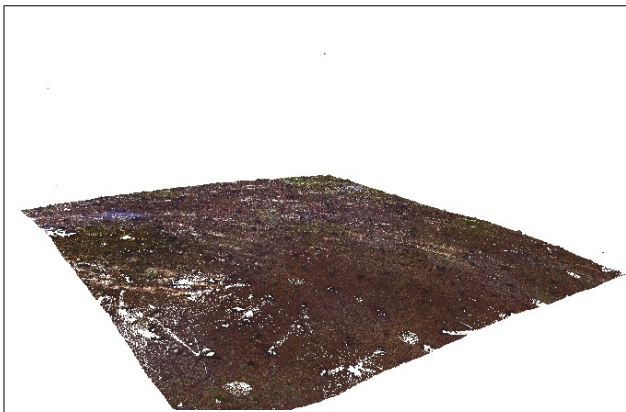
4.1.4 Cross-section

The **Cross-section** is a very important tool in a process of vegetation segmentation, tree identification and tree measurements. The procedure extracts points in a given relative heights above a surface. Optionally, also extents of x and y direction can be defined.



The input surface is usually DTM of a study area. Grid representation of DTM can be derived from a point cloud by the [Vertical_projection](#) module.

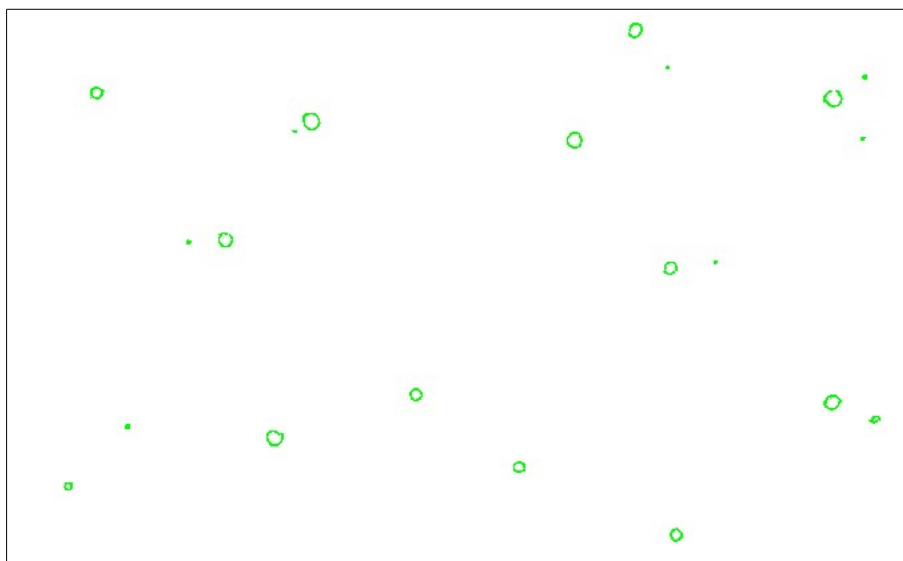
Examples:



DTM points
(cross-section from -0.1 to 0.3 above DTM)



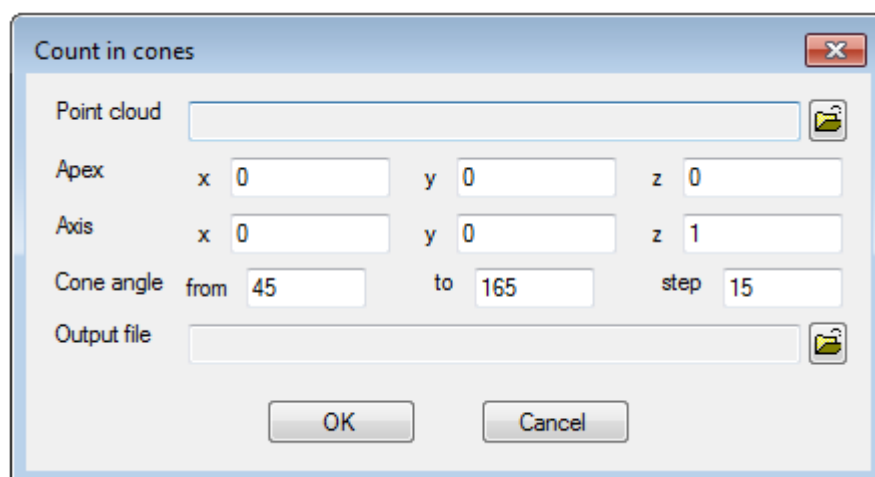
Vegetation points
(cross-section from 0.3 to 99.0 above DTM)



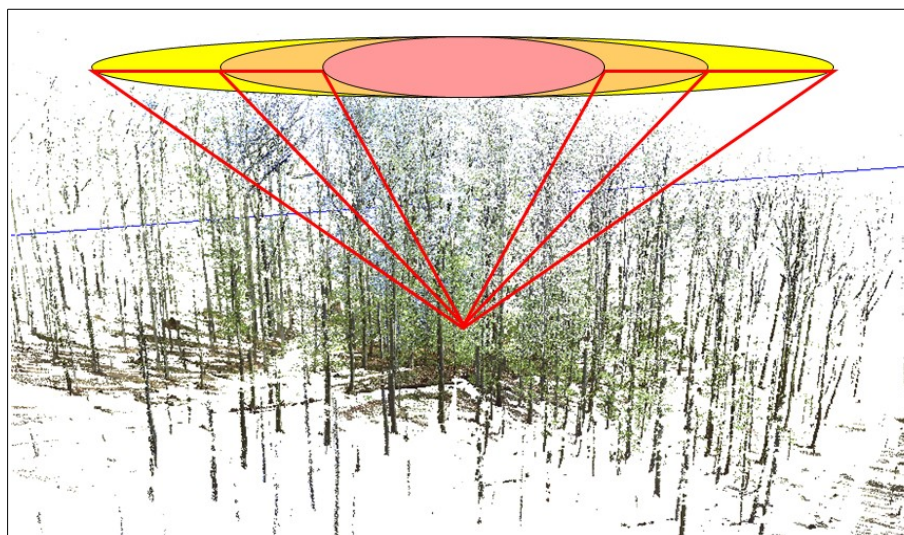
Cross-section of a point cloud at breast height
(cross-section from 1.28 to 1.32 above DTM)

4.1.5 Count in cones

The module **Count in cones** estimates number of reflected laser beams in a set of cones. The cones are defined by position of their apex and axis orientation. The set of cones is generated with the same apex and orientation, but different cone angles. The standard orientation of cones is an upward, so they are oriented on crowns of trees.



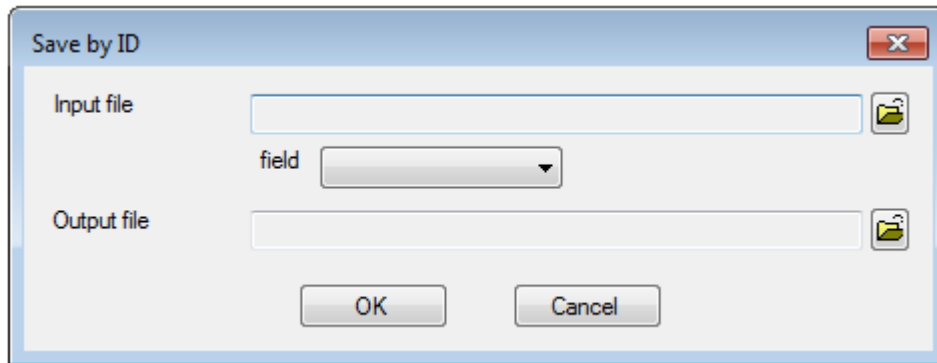
The utility is designed for studies of leaf area estimation and phenological observations. The output data can be used for evaluation of changes in number of laser beam returns from crowns over time.



Standard orientation of cones

4.1.6 Save by ID

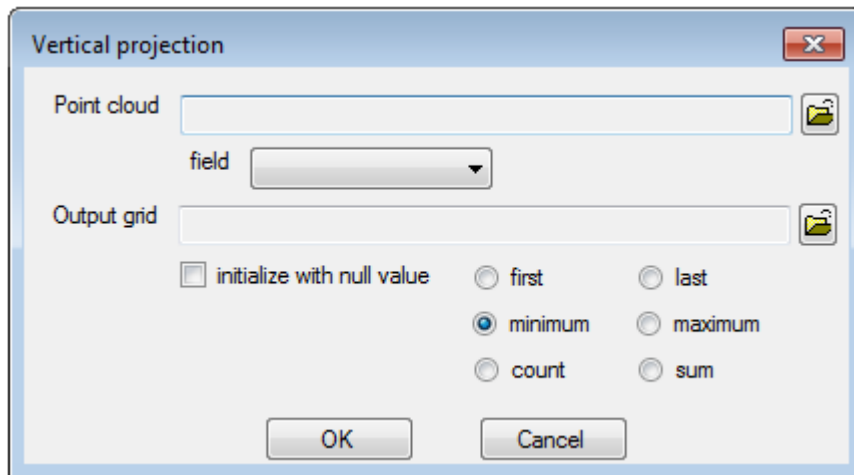
The utility **Save by ID** is used to split an input point cloud to groups. Groups are identified by a unique number stored in the group identification field (e.g. the field *trunk_id*). Each group is saved into a separate output file. The name of the group output file is composed from a given file name and the group identifier.



4.2 Digital terrain model

4.2.1 Vertical projection

The module **Vertical projections** vertically projects any attribute of a point cloud to a two-dimensional output grid. The algorithm calculates value of aggregation function from all points which fall inside a cell. The output grid must be created before running the **Vertical projection** module. The output grid can be created by the utility [Create new grid](#) or the utility [2D grid from point cloud](#).



The aggregation functions:

- **first**: the value of the first point inside a cell,
- **last**: the value of the last point inside a cell,
- **minimum**: the minimum value of points inside a cell,
- **maximum**: the maximum value of points inside a cell,
- **count**: the number of points which fall inside a cell,
- **sum**: the total sum of values of points which fall inside a cell is calculated.

Optionally, the output grid can be initialized by the *null* value. The option should be used during experiments with vertical projection, when the output grid is used several times.

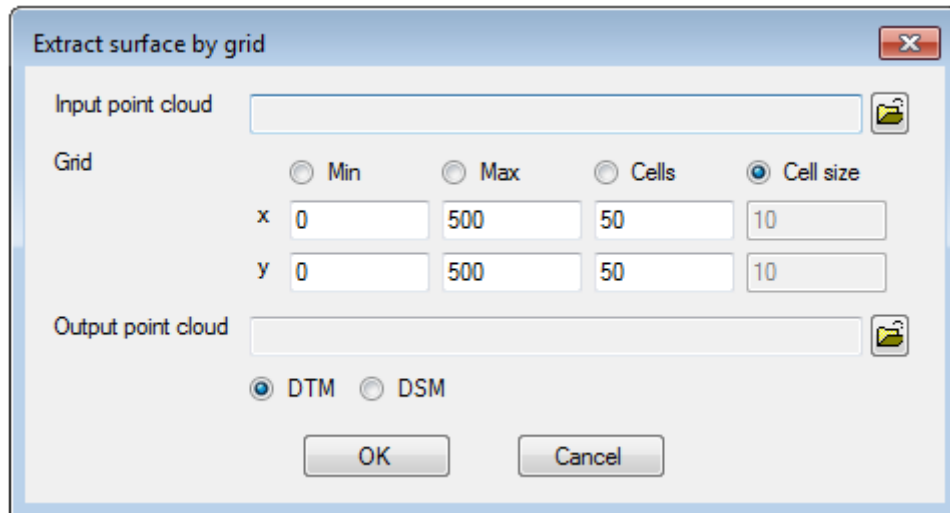
The vertical projection is a very simple and fast method to derive DTM or DSM from a point cloud. The *minimum* aggregation function applied on the z-field of a point cloud is used in process of creation DTM. DSM can be created by application of the *maximum* aggregation function on the z-field.

A density of a scan area coverage by a point cloud can be calculated by use of the *count* aggregation function (the calculated number of points in cell is divided by a cell area) on any point cloud's field.

In some situations a scanned area is not fully covered by a point cloud and there are no points in some grid cells. The values of DTM (DSM) derived from the incomplete point cloud by the vertical projection method are empty in such grid cells. An appropriate gap fill algorithm can be used to fill empty cells, or surface can be interpolated from points generated by the [extract surface](#) module.

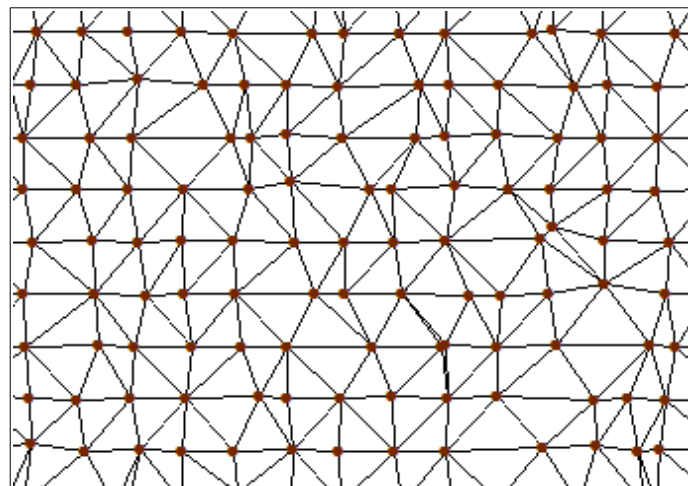
4.2.2 Extract surface

The vertical projection method estimates DTM by finding the minimum z-value for all points within a given cell of a grid. This value is assigned to the cell, so the information about exact position of the minimum (maximum) point is not used. The **Extract surface** module extracts the point with minimum or (maximum) z-value for each cell of a given grid. The points with extremal z-values are saved to an output point cloud. The output point cloud can be exported to a GIS program and used to interpolate DTM or DSM.



The surface interpolated from the extracted points fits well to the terrain (or to the canopy surface), because exact position of points is used in interpolation. Moreover, a spatial interpolation also calculates values for the grid cells, which are not covered by the point cloud.

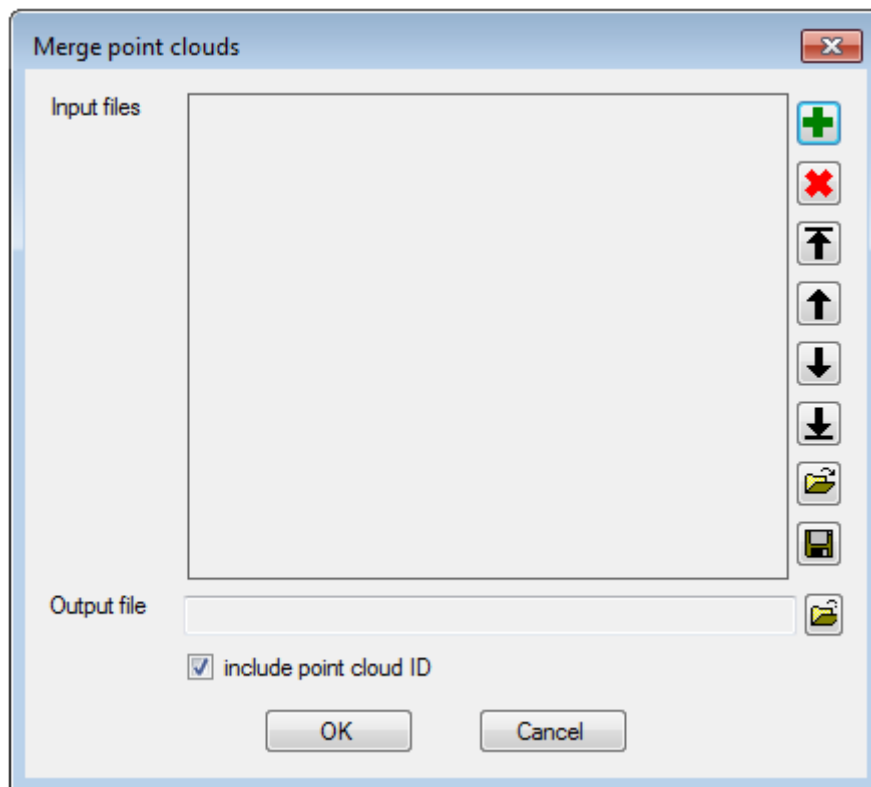
Example:



4.3 Merge point clouds

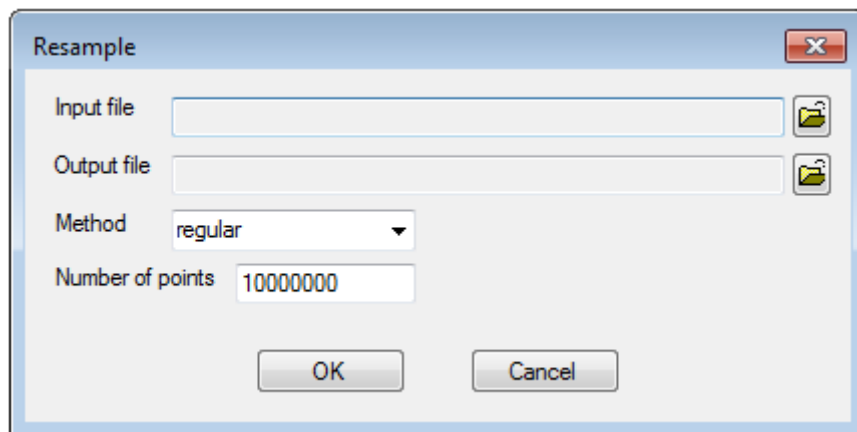
The **Merge** is a useful tool for experiments with point cloud data. The original point cloud can be split into several parts, each of them can be processed in a different way, and results are merged into a final point cloud.

Identifiers of source point clouds can be added to the merged point cloud. The identifier is used to determine a source of points in subsequent analyses and processing. For example, multi-scan of a research plot can be merged from individual point clouds. The merged point cloud can be used for tree identification and measurement in multi-scan mode. Spatial clusters of a cross-section of the merged point cloud can be resampled to simulate single-scan mode.



4.4 Resample point cloud

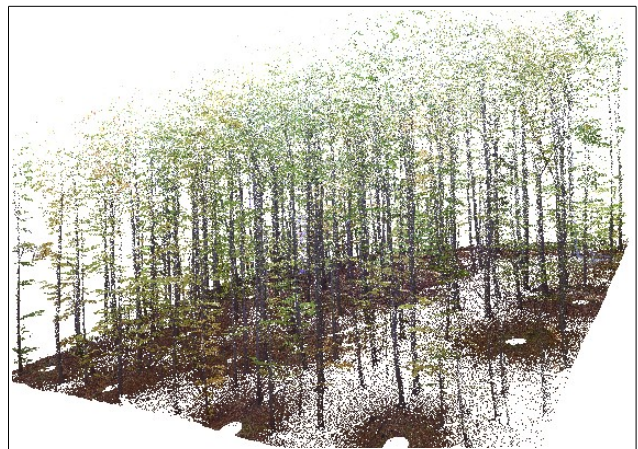
The utility **Resample** extracts a subset of points from a given point cloud. The algorithm selects points in equal intervals from the input point cloud. A given number of points are written to an output file.



Example



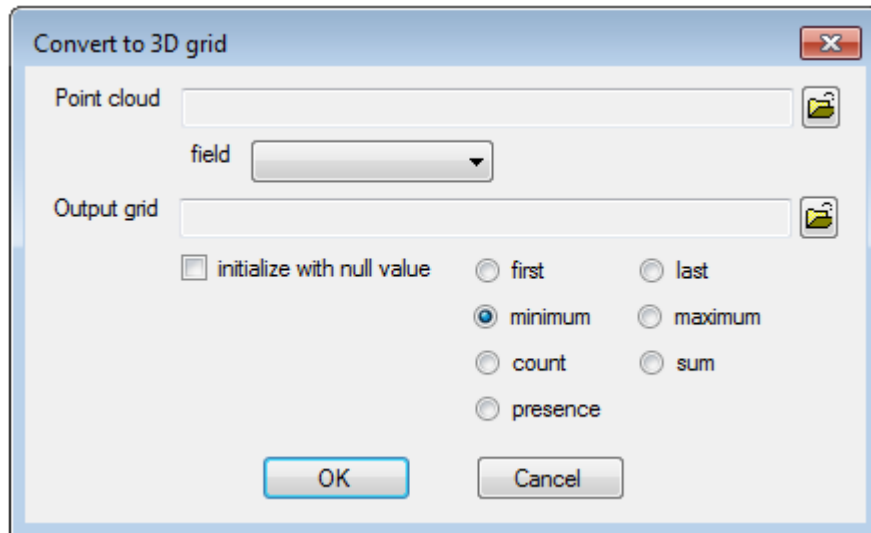
40 millions of points



1 million of points

4.5 Convert to 3D grid

The module **Convert to 3D grid** assigns values of a given point cloud's field to corresponding cells of a three-dimensional output grid. The procedure calculates value of aggregation function from all points which fall inside a cell. The output grid must be created before running the module. The output grid can be created by the utility [Create new grid](#) or the utility [3D grid from point cloud](#).



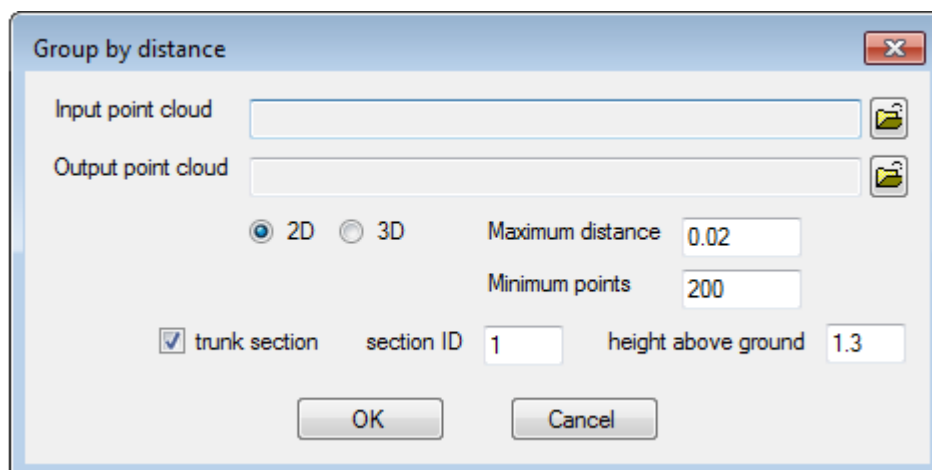
The aggregation functions:

- **first**: the value of the first point inside a cell,
- **last**: the value of the last point inside a cell,
- **minimum**: the minimum value of points inside a cell,
- **maximum**: the maximum value of points inside a cell,
- **count**: the number of points which fall inside a cell,
- **sum**: the total sum of values of points which fall inside a cell is calculated,
- **presence**: 1 is assigned to the output cells, which contain some point. Values of the other cells remain unchanged.

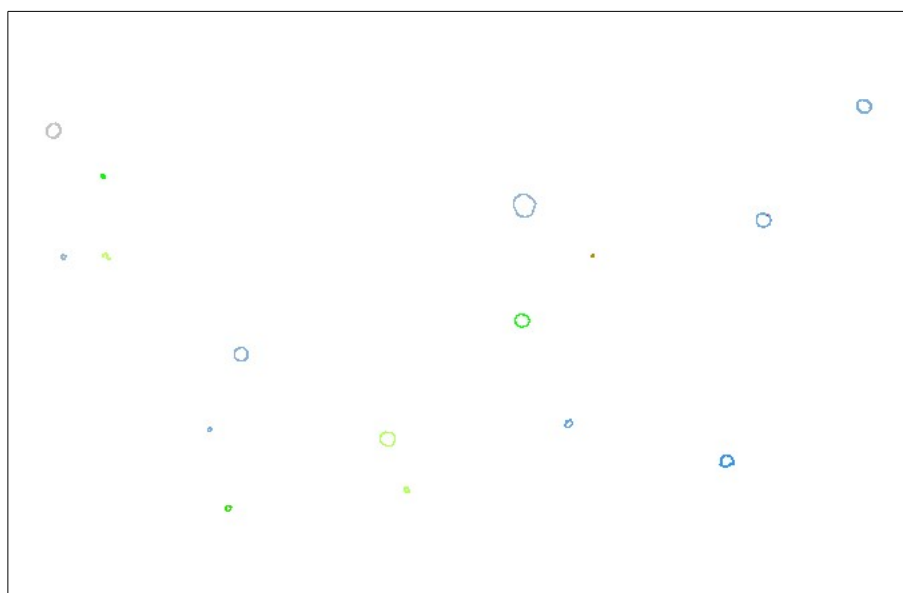
Optionally, the output grid can be initialized by the *null* value. The option should be used during experiments, when the output grid is used several times.

4.6 Group by distance

Spatial clustering is an important step in the process of tree identification and diameter measurement. It is usually applied on a [horizontal cross-section](#) of a point cloud. A point cloud with spatially grouped points is a main input to [tree diameter estimation procedures](#).



Points of the input point cloud are assigned to spatial clusters. A point falls to a group if and only if the distance of the point from the group is less or equal to a given threshold. The same group identification number is assigned to all points of a spatial cluster. The groups with number of points less than a given minimum limit are discarded from an output point cloud. The distance between points can be calculated in 2D or 3D space. Two-dimensional clustering is preferred in grouping points, which represent tree stem cross-sections. If the output is used in tree diameter estimation procedures, check the option “is a trunk section”. The fields *trunk_id*, *sec_id* and *sec_h* will be added to the output point cloud.



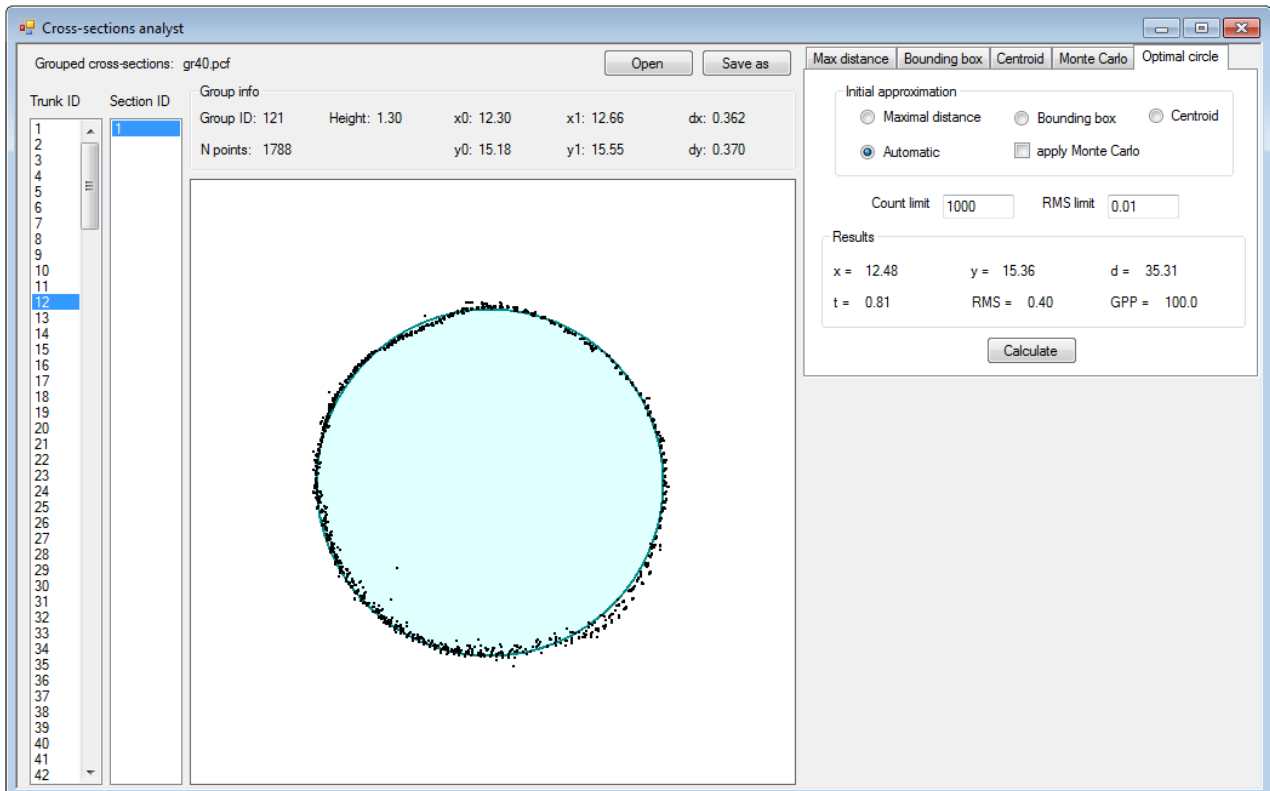
Spatial clusters

The spatial clusters can be further analyzed by the [Cross_sections_analyst](#).

5 Tree identification and measurement

5.1 Cross-sections analyst

Algorithms of tree diameter estimation implemented in the **DedroCloud** software are based on spatial clusters of point cloud cross-sections. A result of tree diameter estimation is strongly affected by density and distribution of points in a cross-section. The **Cross-section analyst** is a complex tool for cross-sections visualisation, interactive tree diameter estimation, and interactive cleaning off cross-sections.



The input of the **Cross-sections analyst** is a point cloud, which contains spatial clusters of tree's cross-sections. The input point cloud must contain the trunk identifier field (*trunk_id*), the section identifier field (*sec_id*), the section height field (*sec_h*), and the spatial cluster identifier field (*group_id*). The input point cloud can be created by the [Group by distance](#) procedure from horizontal [cross-section](#) of a point cloud. The output of the [multi cross-sections](#) module can be analysed too.

Trunks' identifiers are listed in the left most control of the **Cross-section analyst** (*Trunk ID*). When user selects a tree identifier from the list, the list of cross-sections of the selected trunk is displayed in the next control (*Section ID*). The selected cross-section is drawn on the central canvas. Basic information about the selected and displayed cross-section are printed above the cross-section view:

- spatial cluster identifier (*Group ID*),
- number of points of the cross-section (*N points*),
- section height (*Height*),

- coordinates of the lower-left point (x_0 , y_0),
- coordinates of the upper-right point (x_1 , y_1),
- width (dx),
- height (dy).



Due to limited computer speed only 5 000 points of the selected cross-section are drawn.

Position and diameter of the selected cross-section can be interactively calculated by different methods. Parameters and results of diameter estimation methods are listed on the separate tabs:

- coordinates of the cross-section's centre (x , y),
- computation time (t , in milliseconds),
- estimation of the diameter (d),
- root mean square error of the diameter estimation (RMS),
- estimation of the girth coverage by points (GPP , %).

The selected cross-section's points can be interactively deleted by tools from a context menu of the cross-section view. The context menu is accessed with a right mouse click within the control.

The context menu of the cross-section view:

- **Save image:** the current image is saved to a graphics file (**.png*).
- **Select by circle:** points outside a circle are deleted. The circle is defined by two opposite points interactively placed at its circumference.
- **Delete by rectangle:** points inside a rectangle are deleted. Rectangle is defined by two interactively placed corners.
- **Restore points:** all deleted points of the current cross-section will be restored.
- **All methods extent:** the view extent is set to display results of all diameter estimation methods without shift and change of scale. Use this option, when you need to compare diameter estimation of the cross-section by different methods.
- **Trunk extent:** the view extent is set to common extent of all tree cross-sections. All cross-sections will be displayed at the same position and with the same scale. Use this option, when you need to compare diameter estimation at different heights.

The context menu of the Section ID list:

- **Delete:** the selected cross-section will be removed from the list.

The context menu of the Trunk ID list:

- **Delete:** the selected trunk will be removed from the list.
- **Restore:** the list of trunks will be restored. Only the list of trunks will be restored. The function will not restore deleted cross-sections not points.



The changes can be saved by the button **Save as**. The changes must be saved to a new file. The original input file cannot be overwritten. Positions and diameter estimations calculated in the **Cross-sections analyst** are not saved to the output file.

5.2 Tree diameter estimation

The tree diameter is estimated from a spatial cluster of its cross-section at some height above terrain. Point cloud cross-section and spatial clusters must be created by the module [Cross-section](#) and the [Group by distance](#). The tree diameter estimation procedures calculates also diameters from [multi cross-sections](#) of trees. The spatial clusters of tree cross-sections can be interactively visualized, edited and estimated by the [Cross-sections analyst](#).

Initial procedures of tree diameter estimation derive position and diameter of a given cross-section from its geometrical properties. The ***initial methods of diameter estimation*** include:

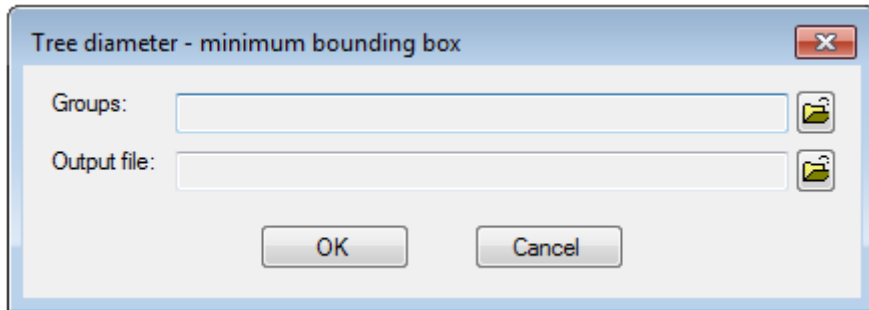
- [the minimum bounding box method](#) (MBB),
- [the centroid method](#) (CEN),
- [the maximum distance method](#) (MDS).

Initial estimations can be improved by refining methods. The refining methods use optimization algorithms to minimize a root mean square error of circle-fitting residuals. The ***refining methods of diameter estimation*** include:

- [the Monte Carlo method](#) (MC),
- [the optimal circle method](#) (OC).

5.2.1 Minimum bounding box method

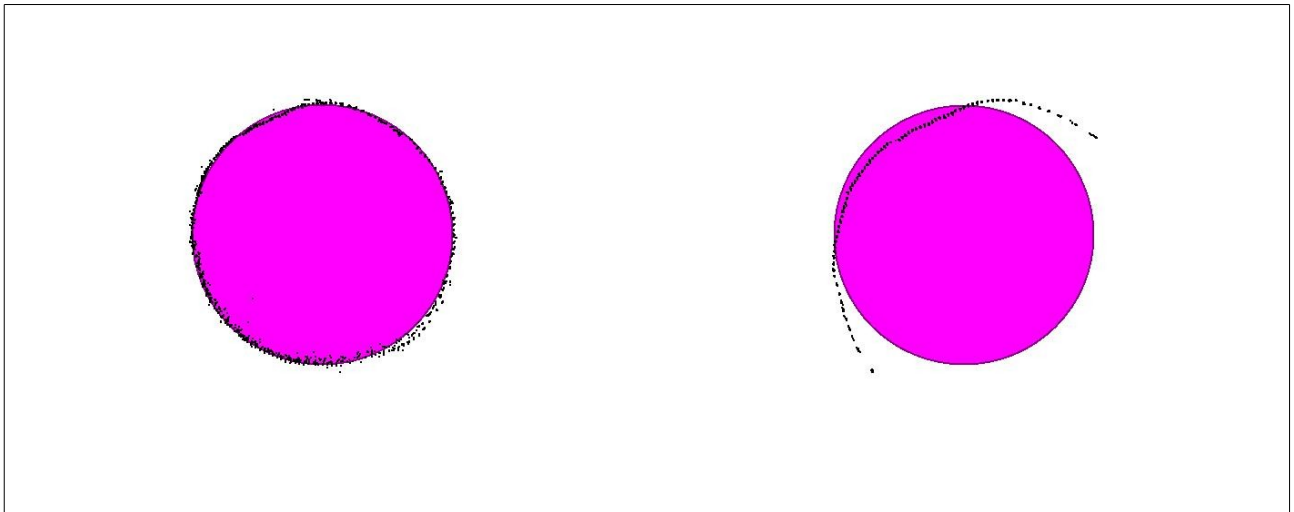
The minimum bounding box method calculates an extent of coordinates in x and y dimension. The extent is described by a minimal two-dimensional box, which contains all points of a cross-section.



The position of a circle is calculated as the centroid of the minimum bounding box, and the radius of the circle is the mean distance of cross-sections' points from the center of the minimum bounding box.

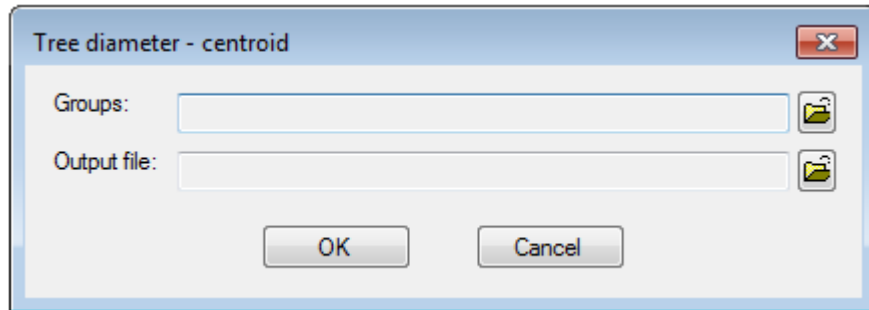
The minimum bounding box provides good estimation of a trunk position and diameter, when a point cloud covers each side of a tree girth. The minimum bounding box of a partly scanned tree is shifted towards a side with higher coverage by points.

Examples



5.2.2 Centroid method

The centroid method calculates a trunk position as an arithmetic mean of a spatial cluster's points. The radius is approximated by the mean distance of points from the trunk's position.



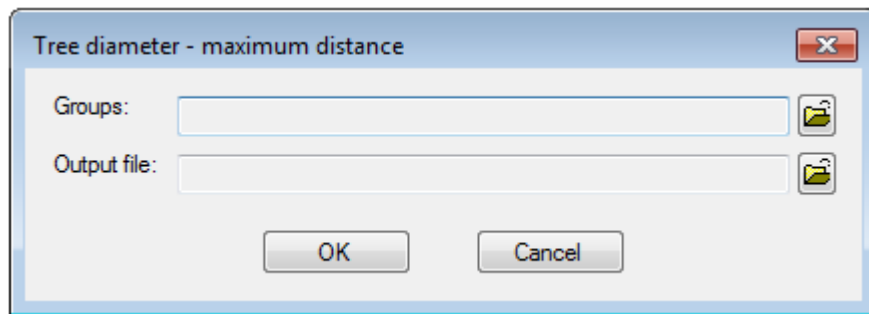
The estimation of position and diameter by the centroid method is strongly affected by a spatial distribution of cross-section's points on the tree's girth.

Examples



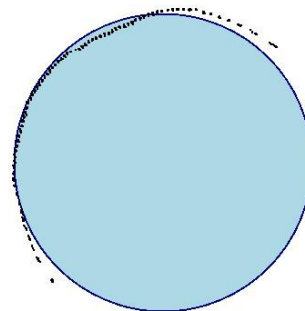
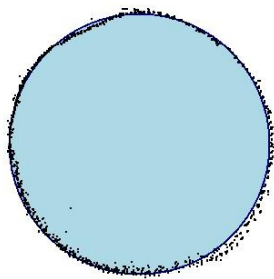
5.2.3 Maximum distance method

The maximum distance method searches a spatial cluster for two most distant points. The position of the trunk cross-section is estimated by the center of the line segment connecting these two most distant points. The radius of the circle is calculated as the arithmetic mean of point distances from the center of the circle.



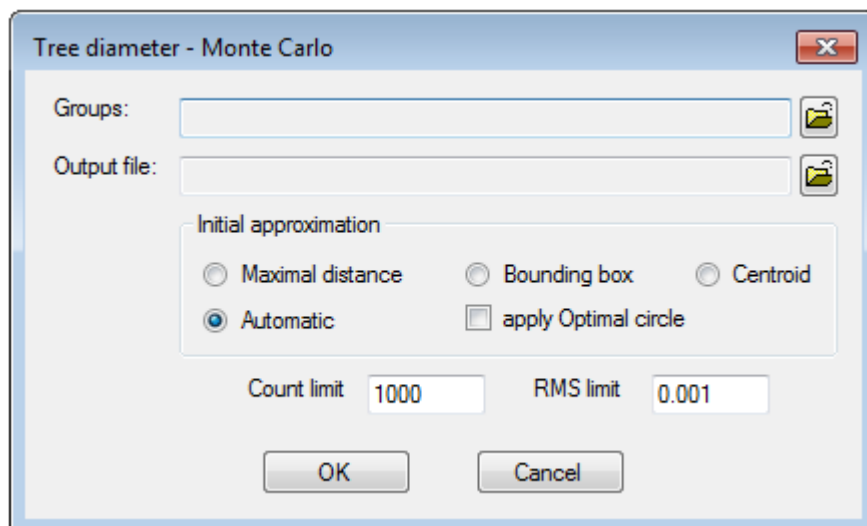
The estimation of cross-section's position and diameter by the maximum distance method is usually very close to the estimations by the refining methods.

Examples:



5.2.4 Monte Carlo method

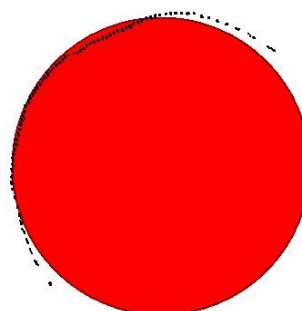
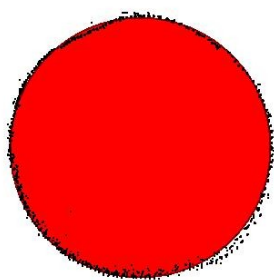
The Monte Carlo method improves the initial estimation of a circle's position and diameter by generating a large number of small shifts in diameter and position of the circle. The circle with the smallest RMS is selected. The calculation will be stopped when the required RMS limit is reached, or after the specified number of simulations.



The initial estimation method can be selected by a user or automatically. The automatic selection calculates approximations by all initial methods and selects the method with the smallest RMS. Optionally, the initial estimation can be improved by the [optimal circle method](#).

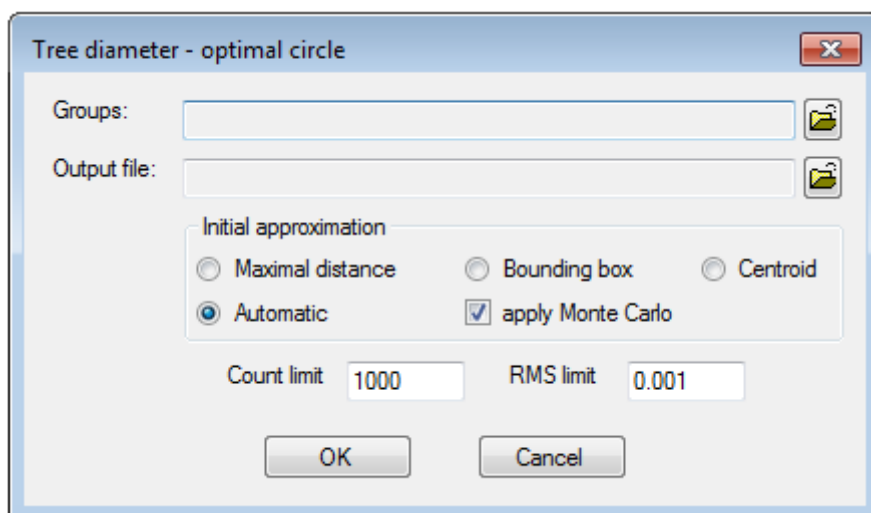
According to our experience, 1 000 repetitions is sufficient to improve accuracy of the initial methods.

Examples:



5.2.5 Optimal circle method

The optimal circle method uses multidimensional mathematical optimization algorithm to find a circle with minimum value of RMS.



The initial estimation method can be selected by a user or automatically. The automatic selection calculates approximations by all initial methods and selects the method with the smallest RMS. Optionally, the initial estimation can be improved by the [Monte Carlo Method](#).

In the most cases, the optimal circle method improves the initial estimation of a cross-section's position and diameter.

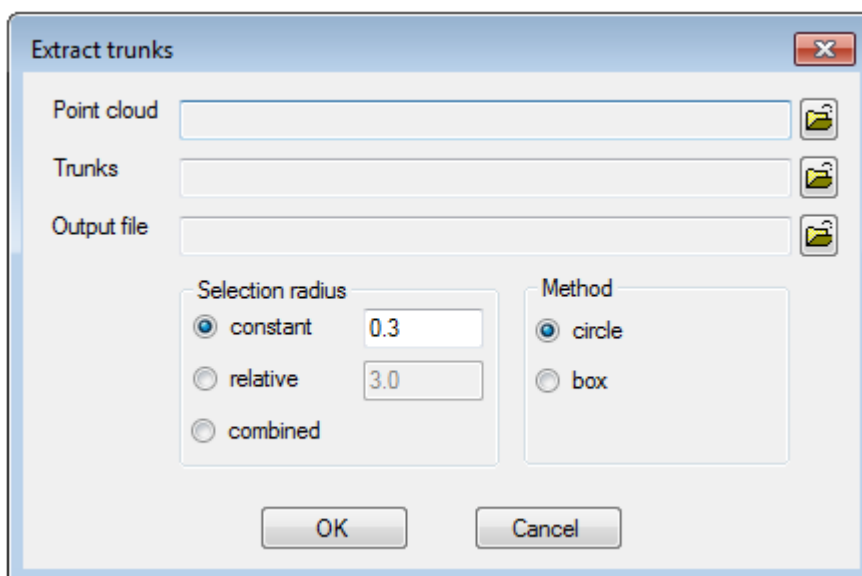
Examples:



5.3 Extract trunks

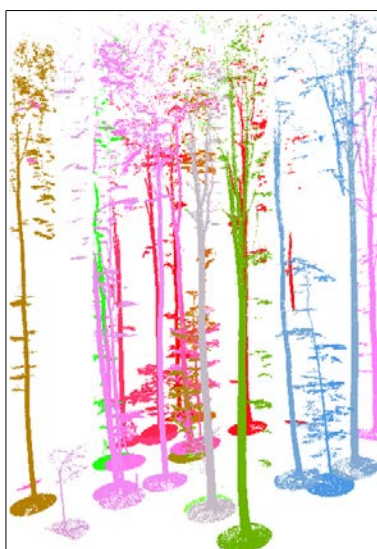
The **Extract trunks** procedure uses a very simple algorithm to assign points to individual trees.

The main input of the procedure is an estimation of tree positions. The approximate trunk position can be calculated by any [tree diameter estimation method](#). The algorithm of the trunk identification generates a vertical cylinder or box and places it at the trunk position. The selection diameter is equal to a given constant for all trees, or is set to a proportion of the tree's diameter estimation (e. g. 3 times larger), or the selection diameter is calculated as the sum of the constant and the proportional enlargement. The trunk's identifier is assigned to all points within the selection object. Only points assigned to some tree are written to an output point cloud. The other points are discarded from the output file.



The extraction of trunks is a computational intensive process. We recommend to [resample](#) the input point cloud to a lower point density before trunk extraction.

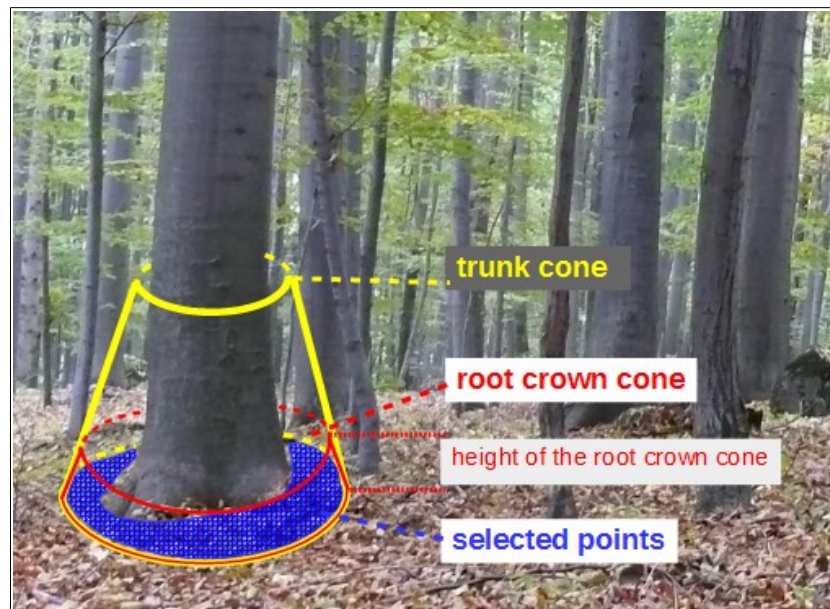
Example:



5.4 Tree base estimation

Approximate positions of trunks can be estimated from a point cloud cross-section. This information can be used to derive elevation of a trunk base from a point cloud data. If elevation of a tree base is known, then more accurate [multiple cross-sections](#) of the tree can be extracted from a point cloud.

Due to an inaccuracy of a tree position, the lower part of the trunk is modelled by a truncated cone. The upper and the lower radius of the truncated cone are derived from the tree diameter estimation. The estimated diameter of the tree is enlarged in order to ensure coverage of the tree's lower part and a nearby terrain by a point cloud. The truncated cone contains points reflected from the nearby terrain as well as points reflected from the trunk. Therefore, only points from a root crown cone are used in the estimation of the tree base elevation. The root crown cone is determined by its height above ground. Finally, in order to eliminate local extremes of the terrain, points reflected from the tree and its root crown only points from a specified lower range of heights are selected and used for the estimation of tree base elevation. The height range is specified as a percentage interval. The elevation of the tree base is calculated as the minimum, maximum or average value of elevation of selected points.



The main input of the **Tree base estimation** procedure is an initial estimation of tree positions and diameters. Any method of [tree diameter estimation](#) can be used for initial approximation of tree positions.

The upper and lower radius of a trunk cone are calculated using the formula

$$r_C = a + b \cdot r_T$$

where

- r_C is a cone radius,
- a is a constant radius enlargement,
- b is a relative radius enlargement,
- r_T is an estimated radius of a tree.

The upper side of the truncated cone is placed at the height of the initial cross-section. The height of the lower side is calculated from the height of the upper side and the approximate height of the tree cross-section.

Tree base estimation

Trunk positions

Point cloud

Output file

Trunk cone

Upper radius
☒ constant
☐ relative
☐ combined

Lower radius
☒ constant
☐ relative
☐ combined

Root crown

Height
☒ constant
☐ relative
☐ combined

Points selection interval (%)
 lower boundary
 upper boundary

Method
☐ minimum ☒ mean ☐ maximum

☐ save statistics

OK Cancel

The height of a root crown cone is calculated using the formula

$$h = a + b \cdot r_T$$

where

- h is a height of a root crown cone,
- a is a constant height,
- b is a multiplier of an estimated tree radius,
- r_T is an estimated radius of a tree.

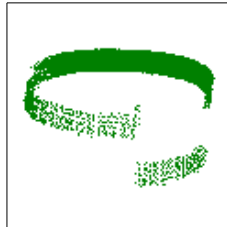
If Save statistics option is checked, minimum, maximum and average z-value of points inside the specified range of the crown root cone are calculated for each trunk.

Results of the tree base estimation are saved to the additional fields of the output file:

- ***base_z***: elevation of a tree base (*float*),
- ***base_minz***: minimum value of a tree base elevation (*float*),
- ***base_maxz***: maximum value of a tree base elevation (*float*),
- ***base_meanz***: mean value of a tree base elevation (*float*).

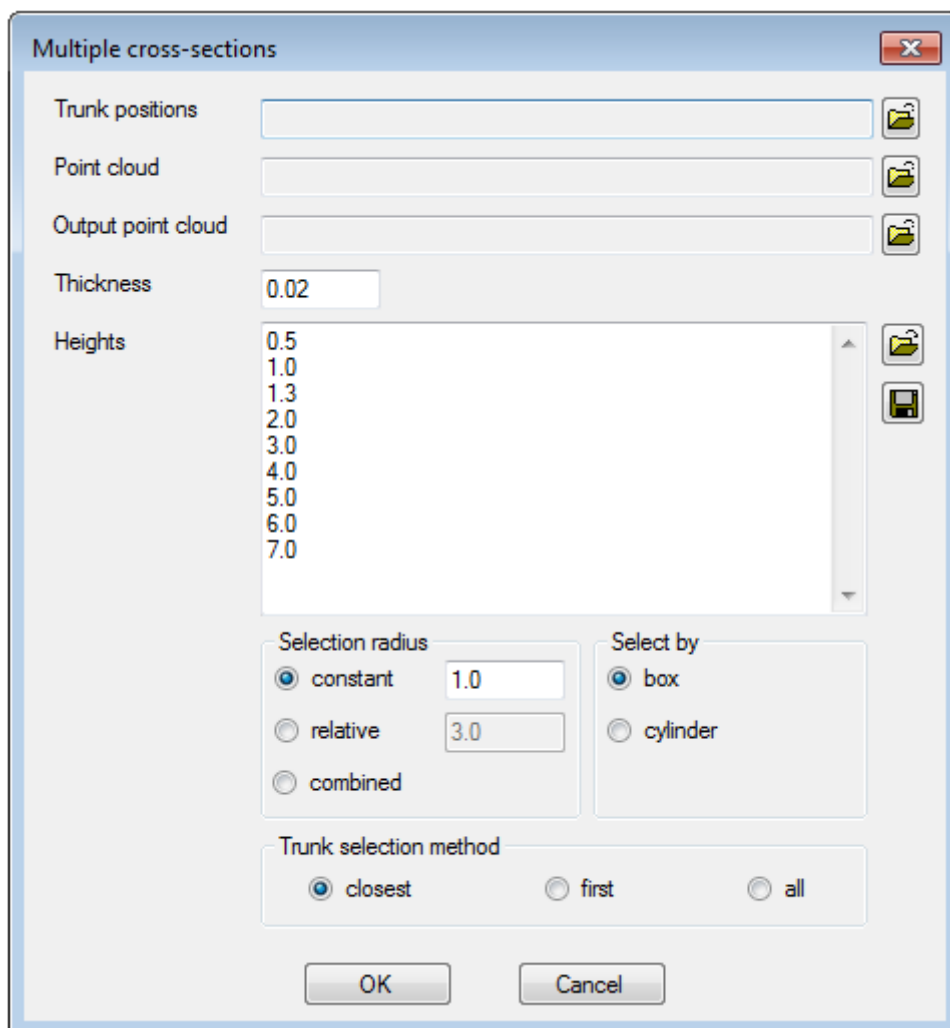
5.5 Multiple cross-sections

Common error of tree cross-sections over a raster DTM is a horizontal shift. If the tree intersects two or more adjacent cells with different elevations, then a cross-section of the tree created by the [Cross_section utility](#) is broken. In this situation, the cross-section consists of several vertically shifted parts. The tree position and diameter estimated from vertically shifted cross-section is not accurate.



Break of the tree cross-section

The **Multiple cross-sections** procedure was designed to eliminate errors of vertical shifts. The height of a tree cross-section is measured from the tree base instead of a raster DTM.



The main input of the multiple cross-section procedure is a point cloud, containing positions, diameters, and base heights of trees.

The first approximation of diameters and positions of trees can be created by from single cross-sections at appropriate height above ground by the [Cross-section](#) procedure. Then base heights are assigned to trees by the [Tree base estimation](#) utility. The output file of the tree base estimation utility contains estimated positions, diameters and base elevations of trees.

The list of cross-sections' heights above tree base can be edited by a user. The list of the heights can be saved to and loaded from a file.

The points are assigned to cross-sections by a cylinder or a box generated at the given height. The radius of a selection (r_s) is calculated from the estimated radius of the tree:

$$r_s = a + b \cdot r_T$$

where

- a is a constant,
- b is a multiplier of an estimated tree radius,
- r_T is an estimated radius of a tree.

This method of the selection radius calculation can be adjusted to user needs. The selection radius can be constant for all cross-sections and all trees (relative = 0.0), or it can be percentage of tree's diameter (constant = 0.0). The combined determination of the selection radius enables user to specify the minimum selection radius (constant) and the variable part specified by the percentage of the tree's radius.

The length of the selection box is equal to the diameter of the selection cylinder. The height of the selection cylinder or the selection box is given by a thickness parameter.

Selection of points by the box is faster than the selection of points by the cylinder.

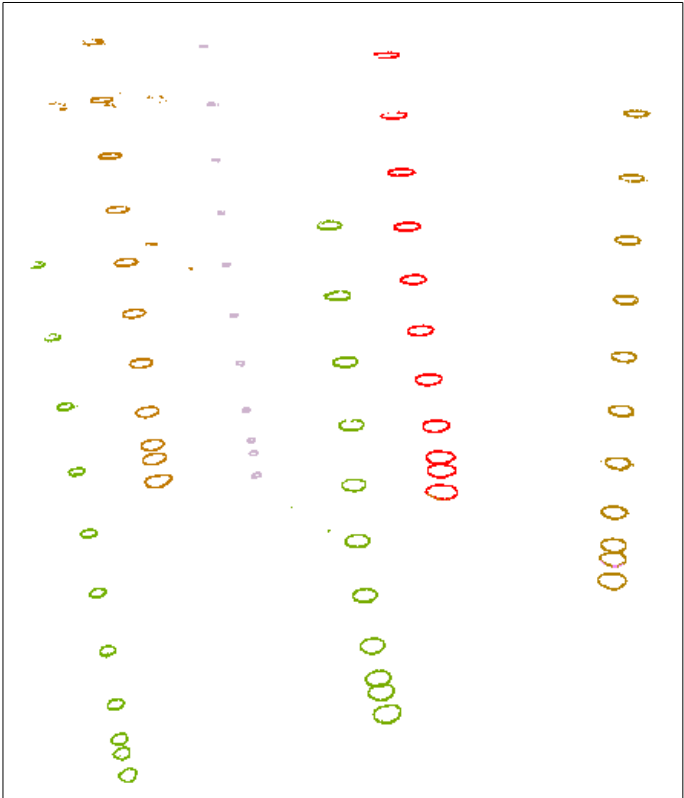
Some points can be within selection cylinders (boxes) of several trees. In this case, point can be assigned only the closest tree, or to the first tree in the input file, or to all trees within the selection radius.

The same trunk identifier is assigned to all cross-sections derived from the same tree estimation. The cross-sections are numbered according the heights. Identifier 1 is assigned to all cross-sections at the first given height above tree base. Identifier 2 is assigned to all cross-sections at the second given height, etc.

Spatial clusters of points are automatically created for each cross-section and each tree. Only points assigned to some cross-section are written to an output file. Fields of the output file:

- **trunk_id**: identifier of a trunk,
- **sec_id**: identifier of a cross-section,
- **sec_h**: height above tree base of a section,
- **base_z**: elevation of a tree base,
- **group_id**: unique identifier of a spatial cluster of points.

Example of the multiple cross-sections:



6 Visualization

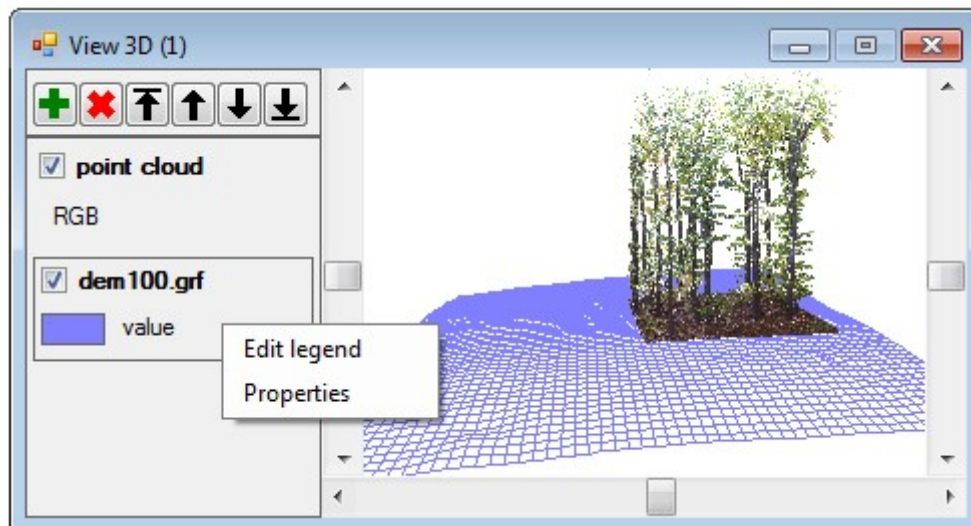
6.1 3D View

Visualization tools are still under development. Visualization capabilities are limited to 3D view of point clouds and 2D grids.

The 3D window consists of a legend panel and a 3D canvas. The legend panel contains a list of themes. A theme can be added, removed, or moved by buttons at the top of the legend panel.

Visibility of a theme is toggled by the check box next to its name. When the theme visibility is turned on, the build of 3D graphics model is started. The build of the 3D model is a time consuming process, so it is run at background. While creating the 3D model, the theme is an inactive. The theme is automatically activated when the 3D model is ready.

Dialogs for a setup of a theme's properties and legend can be started from the theme's context menu. The context menu is activated by a right mouse click on the theme.



Context menu of the 2D grid

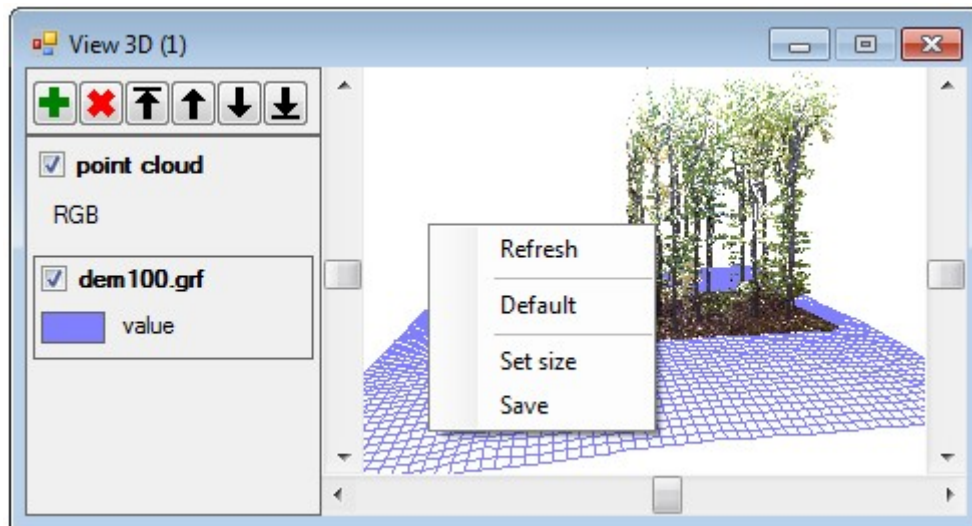
The 3D view controls:

- **rotation**: the horizontal and the vertical scroll bars,
- **zoom**: the mouse's wheel,
- **pan**: the keyboard arrows up, down, left, right (a small step with CTRL key, a large step with SHIFT key),
- **context menu**: the right mouse click inside the 3D canvas.

The 3D canvas context menu:

- **Refresh**: redraw then view.
- **Default view**: set view to the full extent of the visible layers.

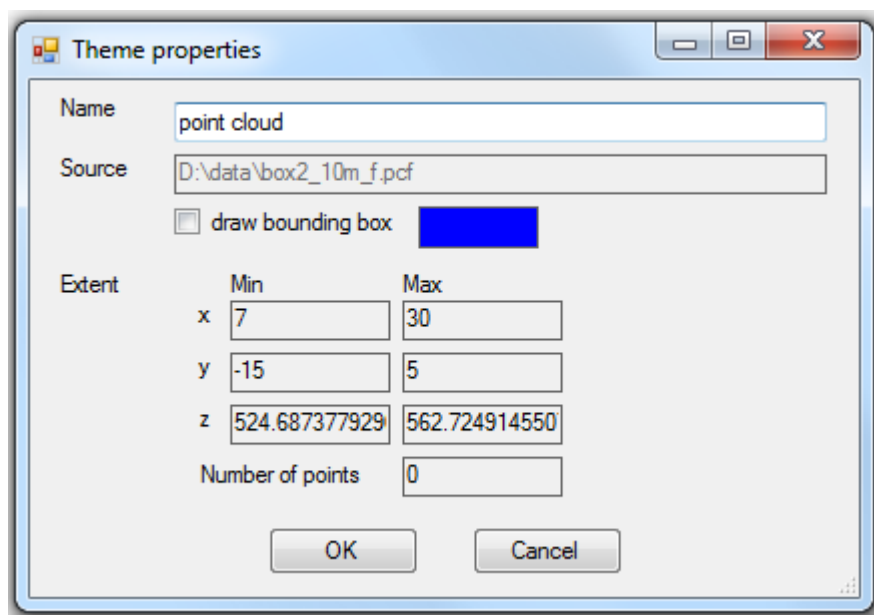
- **Set size:** set the width and the height of the 3D view area. The exact width and height of the canvas.
- **Save:** save the current view to a graphic file (*.jpg). The size of the output image corresponds to the size of the 3D canvas.



Context menu of the 3D canvas

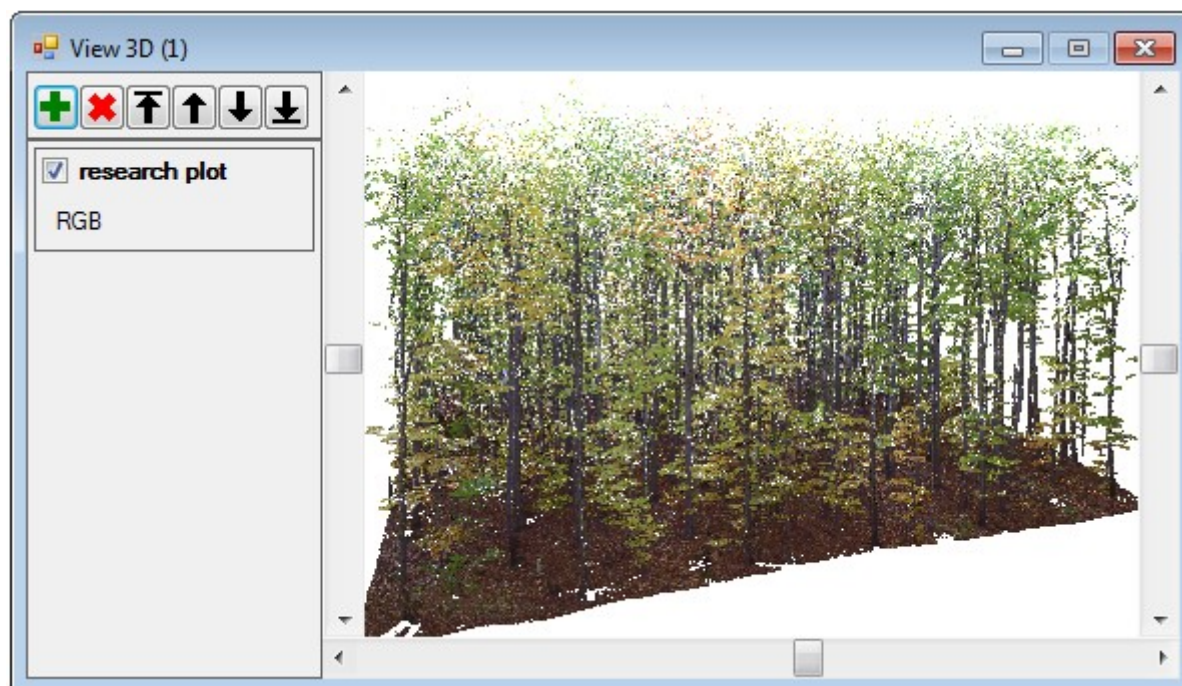
6.2 Point cloud theme

A point cloud is displayed as a subset of its points. A size of the point subset is limited by the *Pcd.ThemeMaxPointsM* parameter in the DendroCloud's [configuration file](#). The number of points should be set in accordance to the performance of a used graphics card.

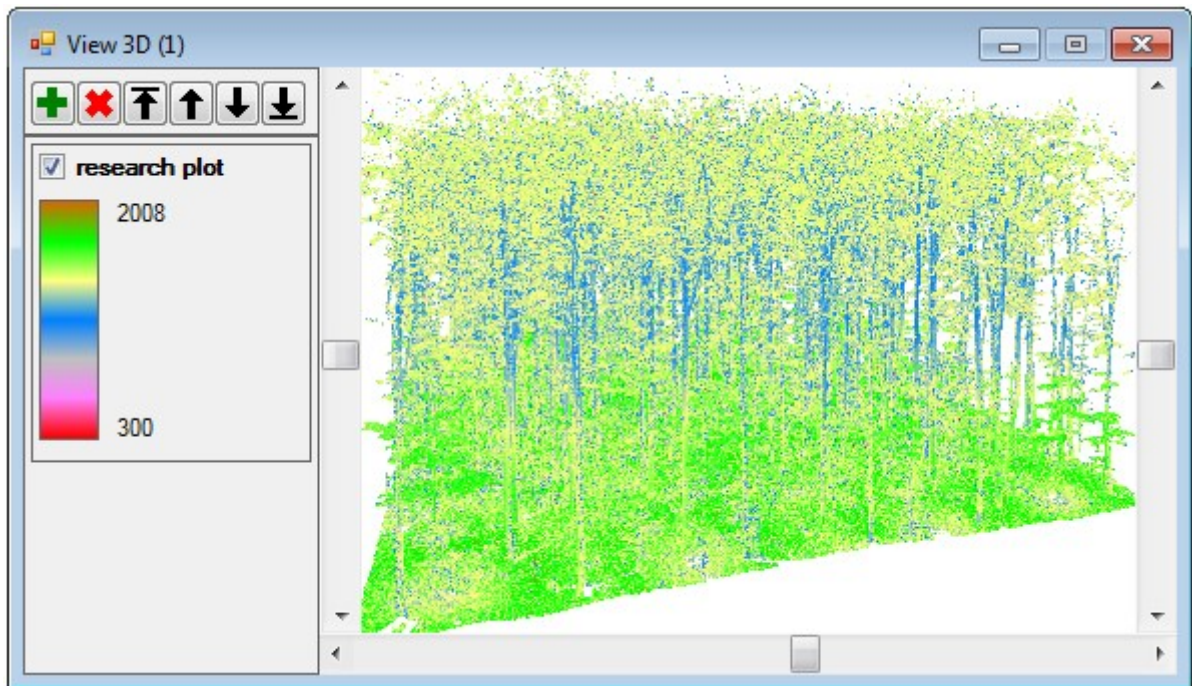


Properties of the point cloud theme

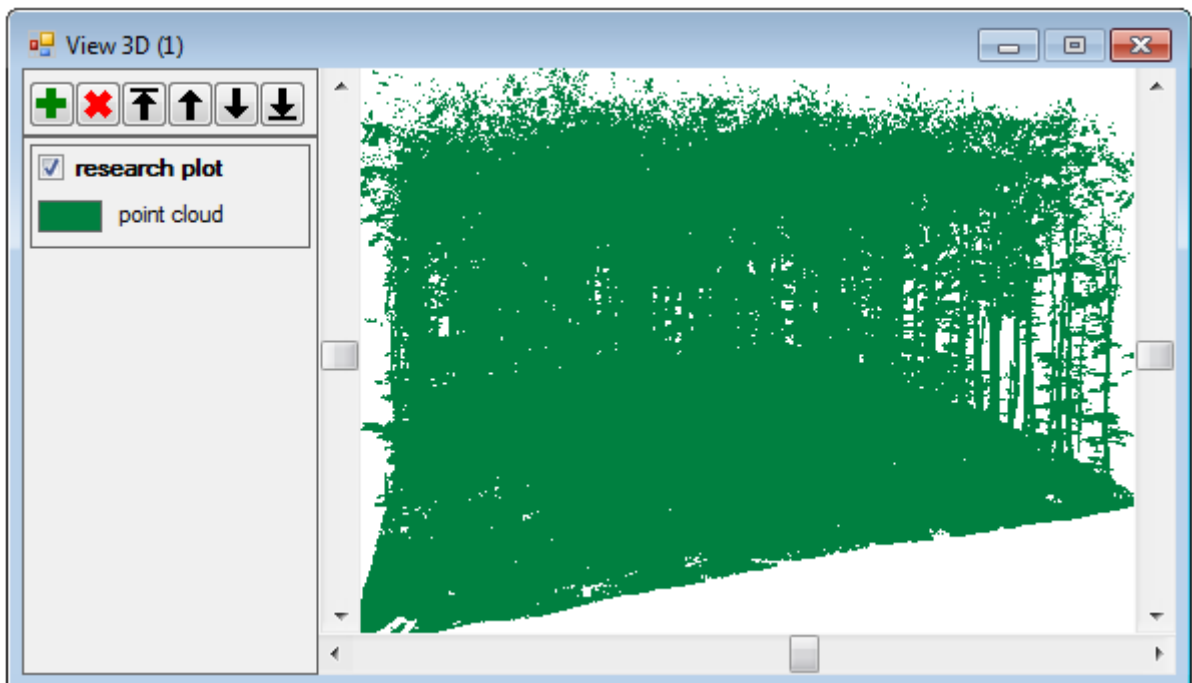
A name of a point cloud theme in the content panel can be set in the Properties form.



True colour (RGB) legend of the point cloud



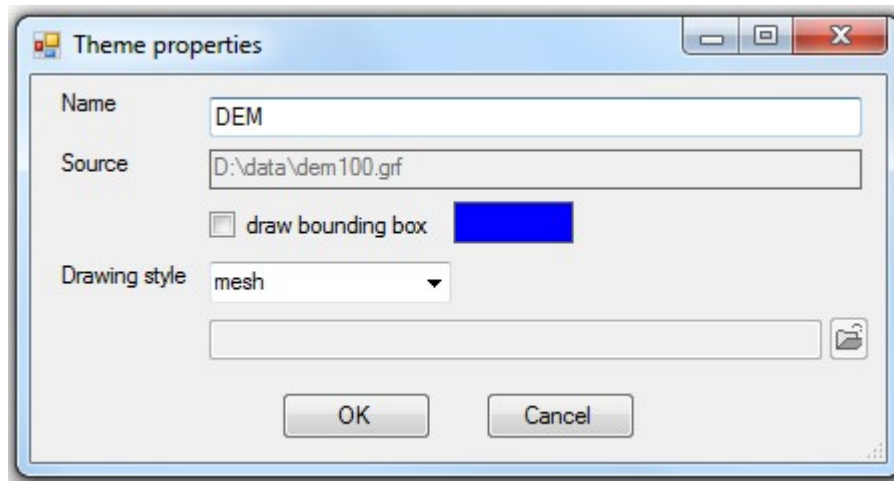
Continuous legend applied on the intensity field



Single symbol legend of the point cloud

6.3 2D grid theme

A 2D grid theme usually represents digital elevation model or digital surface model, which was derived from a point cloud.

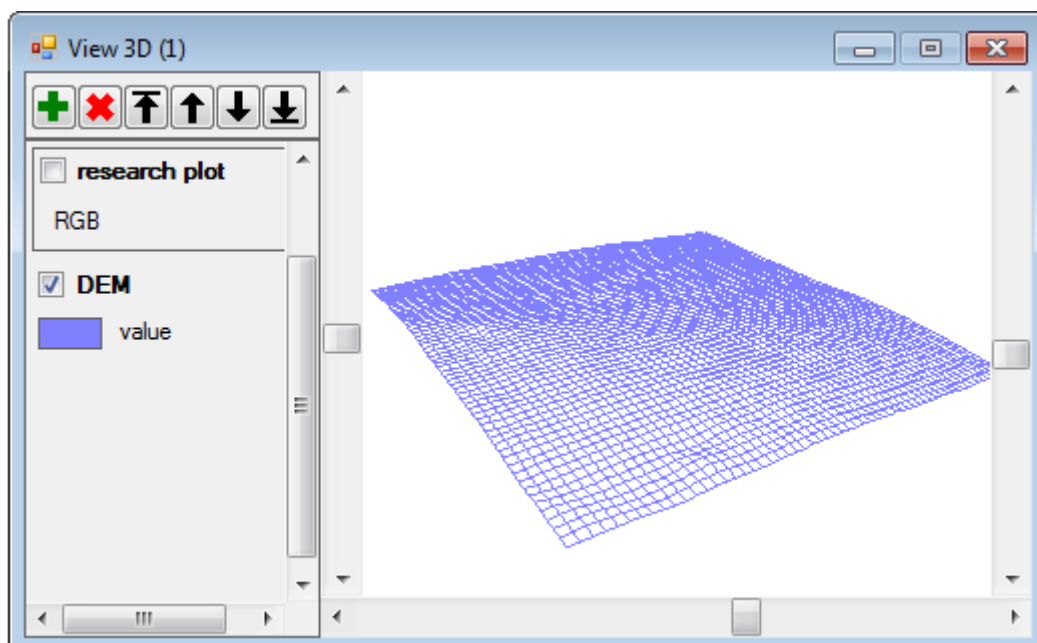


Properties of the 2D grid

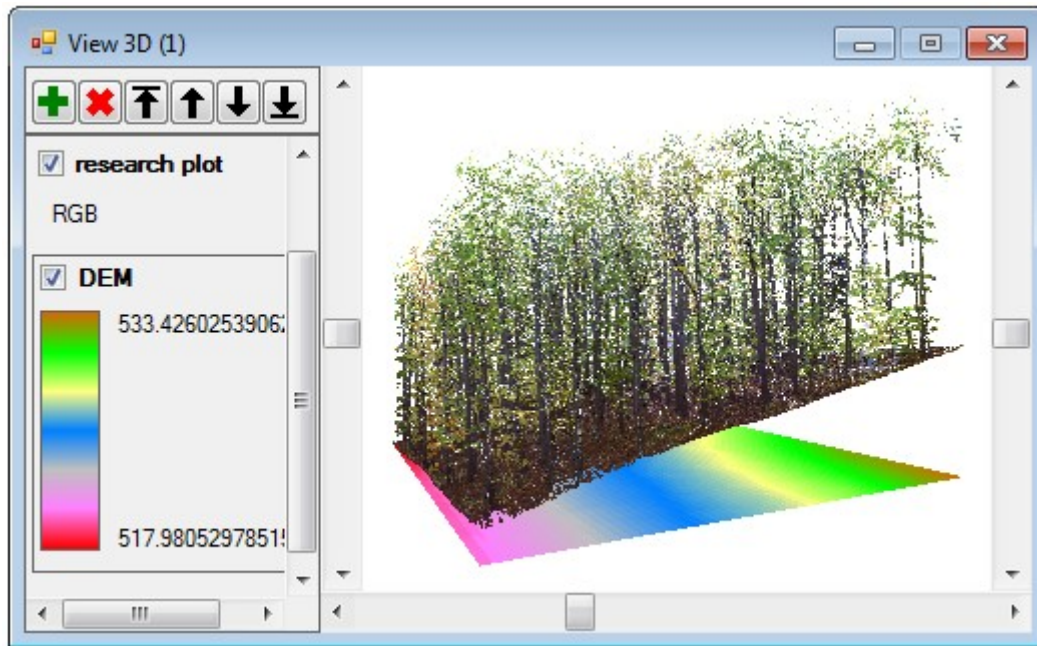
The name of the 2D grid theme in the content panel can be set in the Properties form.

Three drawing styles of a 2D grid are supported: plane, mesh, and blocks.

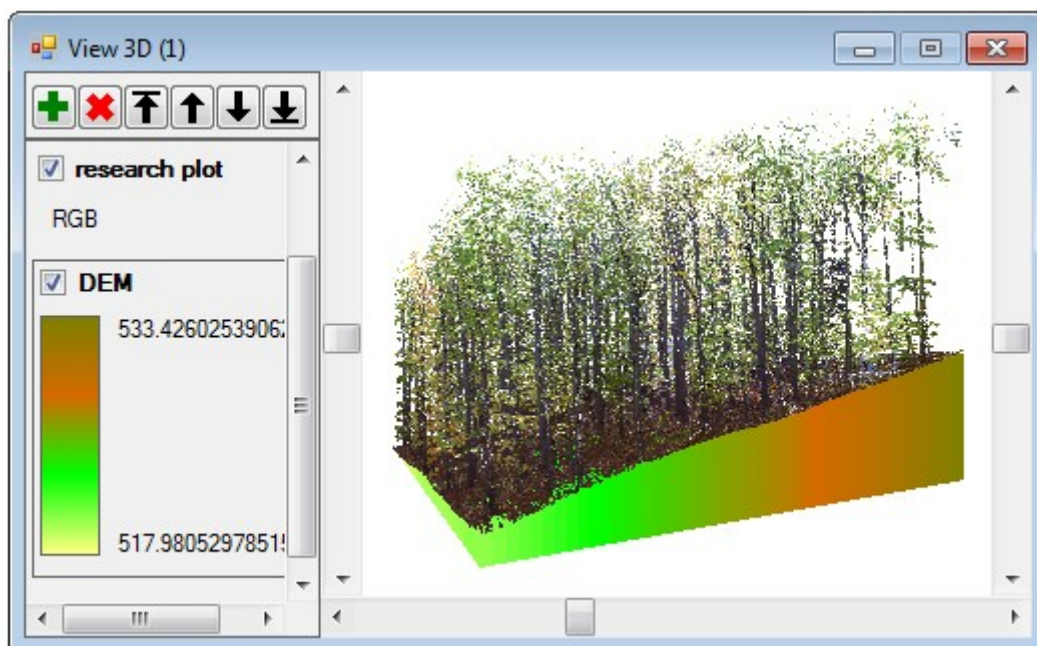
Colour can be changed by mouse click on the drawing symbol in the contents panel.



Mesh drawing style of the 2D grid



Plane drawing style of the 2D grid



Blocks drawing style of the 2D grid

List of abbreviations

| | |
|------|---|
| CEN | Centroid method of tree diameter estimation |
| CHM | Canopy Height Model |
| CRP | Close-range Photogrammetry |
| CSV | Comma Separated Value file |
| DBH | Diameter at Breast Height |
| DSM | Digital Surface Model |
| DTM | Digital Terrain Model |
| GIS | Geographical Information System |
| GPP | Girth cover by Point cloud in Percentage |
| ID | Identifier |
| LAS | LASer File Format |
| MC | Monte Carlo method of tree diameter estimation |
| MBB | Minimum Bounding Box method of tree diameter estimation |
| MDS | Maximum Distance method of tree diameter estimation |
| MLS | Mobile Laser Scanning |
| OC | Optimal Circle method of tree diameter estimation |
| TLS | Terrestrial Laser Scanning |
| VRML | Virtual Reality Modelling Language |

Appendix A: Data types

| Code | Data type | C# | Size (in bytes) | Range |
|------|------------------------|--------|--------------------|--|
| 0 | unsigned byte | byte | 1 | 0 255 |
| 1 | byte | sbyte | 1 | -128 +127 |
| 2 | unsigned short integer | ushort | 2 | 0 65,535 |
| 3 | short integer | short | 2 | -32 768 +32 767 |
| 4 | unsigned integer | uint | 4 | 0 4 294 967 295 |
| 5 | integer | int | 4 | -2 147 483 648 +2 147 483 647 |
| 6 | unsigned long integer | ulong | 8 | 0 18 446 744 073 709 551 615 |
| 7 | long integer | long | 8 | -9 223 372 036 854 775 808 +9 223 372 036 854 775 807 |
| 8 | float | float | 4 | $-3.402823 \cdot 10^{38}$ $+3.402823 \cdot 10^{38}$ |
| 9 | double | double | 8 | $-1.79769313486232 \cdot 10^{308}$ $+1.79769313486232 \cdot 10^{308}$ |
| 255 | unknown | - | - | - |

Appendix B: Extended ASCII raster format

The extended ASCII raster format is compatible to ESRI raster ASCII format (www.esri.com). Additional meta-information is included in the header of the raster interchange file, which determines size and position of a raster in z and t dimensions. A text file can be in 2D, 3D or 3DT format.

Standard parameters of the ESRI raster ASCII format header (2D):

- ***ncols***: number of columns,
- ***nrows***: number of rows,
- ***xllcorner***: x-coordinate of the centre of the lower-left grid's cell,
- ***yllcorner***: y-coordinate of the centre of the lower-left grid's cell,
- ***xcellsize***: cell width,
- ***ycellsize***: cell depth,
- ***nodata_value***: null value, default -9999.

Additional parameters of the extended ASCII raster format header for z-dimension (3D):

- ***nlays***: number of layers,
- ***zllcorner***: x-coordinate of the centre of the lower-left grid's cell,
- ***zcellsize***: cell height.

Additional parameters of the extended ASCII raster format header for t-dimension (3DT):

- ***nticks***: number of ticks.

The raster data are stored by ticks, layers, rows and columns. The first line of the data sections contains the first row of the first layer of the first tick. It is followed by the next rows of the first layer. The second raster layer is stored after the first layer.

The example of the text file in extended ASCII raster format:

```
ncols 116
nrows 116
nlays 1
nticks 1
xllcorner -22
yllcorner -35
zllcorner 314.61
xcellsize 0.5
ycellsize 0.5
zcellsize 0.5
nodata_value -9999
528.00 527.99 528.07 528.12 528.26 528.32 528.39 528.51 528.56 528.63 528.72 528.80 528.88 528.94
529.02 529.08 529.11 529.14 529.25 529.33 529.46 529.59 529.67 529.73 529.74 529.76 529.78 529.85
```

529.91 529.98 530.01 530.08 530.11 530.19 530.21 530.28 530.36 530.37 530.44 530.50 530.55 530.67
530.74 530.75 530.77 530.86 530.94 531.01 531.03 531.10 531.17 531.21 531.24 531.31 531.35

.....

Appendix C: Standard fields of a point cloud

Obligatory fields

| Name | Data type | Description |
|-------------|------------------------|-------------------------|
| x | <i>float or double</i> | x-coordinate of a point |
| y | <i>float or double</i> | y-coordinate of a point |
| z | <i>float or double</i> | z-coordinate of a point |

Optional fields

| Name | Data type | Description |
|-------------|------------------|-----------------------------------|
| r | <i>ubyte</i> | red component of point's colour |
| g | <i>ubyte</i> | green component of point's colour |
| b | <i>ubyte</i> | blue component of point's colour |
| i | <i>int</i> | intensity |

LAS fields

| Name | Data type | Description |
|-------------|------------------|--|
| i | <i>int</i> | intensity |
| rn | <i>int</i> | return number |
| nor | <i>int</i> | number of returns |
| gpstime | <i>double</i> | optional, used in aerial and mobile laser scanning |

Tree and cross-section identification

| Name | Data type | Description |
|-------------|------------------|--|
| group_id | <i>int</i> | unique identifier of a spatial cluster |
| trunk_id | <i>int</i> | identifier of a trunk |
| sec_id | <i>int</i> | identifier of a cross-section |
| sec_h | <i>float</i> | cross-section height above ground or tree base |

Diameter estimation

| Name | Data type | Description |
|-------------|------------------|--|
| sec_r | <i>float</i> | radius of cross-section |
| sec_d | <i>float</i> | diameter of a cross-section |
| sec_n | <i>int</i> | number of points in a cross-section |
| sec_rms | <i>float</i> | RMS of a diameter estimation |
| sec_t | <i>float</i> | calculation time of a diameter estimation (in seconds) |
| sec_gpp | <i>float</i> | girth coverage by points (%) |

Tree base estimation

| Name | Data type | Description |
|-------------|------------------|--|
| base_z | <i>float</i> | elevation of a tree base |
| base_minz | <i>float</i> | minimum value of a tree base elevation |
| base_maxz | <i>float</i> | maximum value of a tree base elevation |
| base_meanz | <i>float</i> | mean value of a tree base elevation |