Digital Photogrammetric System

PHOTOMOD

Version 8.0

USER MANUAL

Neural processing of LIDAR data (Astra Linux 1.7 / Astra Linux 1.8)

RACURS

Table of Contents

1. About

1.1. Purpose of the document

This document describes the functionalities of the software package designed for the neural processing of LIDAR data.

This software package includes the digital photogrammetric system *PHOTOMOD*, as well as an extra module, designed for work in conjunction with *PHOTOMOD*. It is a program used for processing and classifying LIDAR point clouds containing information about point colors.

The module for neural processing of LIDAR data (hereafter the *module* or *program*) includes tools for preparation and generalization of LIDAR data for neural network learning; LIDAR data classification; learning sampling for handling LIDAR data.

The main part of this document includes a description of the software installation and configuration; an information on data processing parameters and sequence; a list of main conventions and terms; a description of input and output data; and a manual for preconfiguring a local network.

1.2. LIDAR data

Lidar (also written LIDAR or LiDAR) is a remote sensing technology that measures distance by illuminating a target with a laser and analyzing the reflected light. Aerial LIDAR systems are represented by laser scanner for remote sensing of the earth's surface. *LIDAR data* – it is data obtained by LIDAR systems, which are installed on aircraft.

LIDAR data (and also some *point clouds*, which can be obtained from another sources) are delivered in LAS format files. The *PHOTOMOD* software supports *ASPRS* (*American Society for Photogrammetry and Remote Sensing*) data format, from version 1.0 till version 1.4. LAS format files include the **X**,**Y** and **Z** coordinates of the laser reflection points and their attributes (see the LAS specification published by *ASPRS*).

Points of the laser reflection from the earth's surface, which coordinates and attributes are contained in the LAS format files, is called the *LIDAR points* in this documentation. To work with LIDAR data and point clouds, use **Terrain › LAS** menu (see the "LIDAR data processing" User Manual).

2. About system

The module for neural processing of LIDAR data is designed to classify 3D point clouds containing information about point colors, regardless of the way they were obtained, by laser scanner or by photogrammetric method (see "Dense DEM generation using

SGM method” in the “DTM Generation” User Manual and “Point cloud coloring by project images” in the “LIDAR data processing” User Manual).

The classification implies the analysis and selection of points from a LIDAR cloud for further assigning one of the classes to these points provided by the system. The system employs the modified (extended) standard LIDAR point classification provided by the applied LIDAR data storage format.

Cars, **furniture** and **wall** LIDAR point classes are used by the program but are not part of the standard *ASPRS* classification. The program employs the “user definable” classes from the standard *ASPRS* classification, treating them as cars and road accessories within the system.

The system allows users to access provided (saved) paths to data files, as well as sets of data processing options (hereafter referred to as *presets*) both pre-installed (supplied with the system) and created (modified) directly by the user.

Some system-provided operations can be carried out in the *distributed processing mode* using standard *PHOTOMOD* tools (see “Distributed processing” in the “General information” User Manual).

Input data for the system are LAS lidar point clouds (`*.las`), that contain information on point colors. Output LAS files contain extra information about the classes assigned to LIDAR points during processing.

PHOTOMOD provides for operator control of processing results and interactive operations (see the “LIDAR data processing” User Manual).

2.1. System requirements

Minimal configuration for workstation:

System component	Recommended configuration
CPU	*Intel Core i7* or present-day *Intel Xeon* (2.8 GHz or equivalent)
Memory (RAM)	32 GB (minimum 16 GB)
Video adapter	*NVidia Tesla A100*, *NVidia GeForce RTX 3060*, *NVidia Quadro T1000* (or better)
SATA (HDD/SSD)	4 TB
Operating system	*Linux* distributions: *Astra Linux 1.8*, *Astra Linux 1.7*

For the security key, an USB port is required (to use local) or network connection (to use system by network).

CPU

It is currently recommended to use systems based on *Intel Core i7* (or better) or present-day *Intel Xeon*, 2.8 GHz or equivalent.

RAM

It is strongly recommended at least 16 GB RAM for stable work in the system. If less, processes could take rather more time.

Video adapter

Work in the program is possible only on PCs with *NVidia* video adapters (8 GB or higher).

⚠ There is no guarantee that the module for neural processing of LIDAR data will work with other graphics adapters.

The greatest computational power of the workstation is needed for the point cloud classifier training process. At present, the maximum optimization of time spending when performing this operation can be achieved using *NVidia Tesla A100* video adapters.

Network adapter

For project processing using network it is recommended to connect server to hub using at least 10 Gbit/s network interface.

Hard disk drive (HDD)

It is recommended to use SSD-devices for the better performance.

⚠ For data management, it's necessary to consider the feature of *Linux* that hard drive partitions, USB drives, network drives, and other data carriers connected to the workstations are to be *mounted*.

3. System installation

Installing the software package on a workstation where the *Racurs* software products have not previously been installed includes the following steps:

1. *PHOTOMOD* (or *PHOTOMOD UAS*) installation including the following operations:
    - *PHOTOMOD* installation;
    - Initial *PHOTOMOD* setup;
    - Disabling all running *PHOTOMOD* units.

PHOTOMOD installation is described in detail in “System installation” in the “General information” User Manual.

PHOTOMOD UAS installation is described in detail in “System installation” in the “Processing of UAS data” User Manual.

The *PHOTOMOD UAS* program is purposed to process data from unmanned aircraft system (UAS). There is a limit on using of the source data when working with program. Only central projection images with size not more than 100 Mpix could be used a source data.

The *PHOTOMOD UAS* program is a stand-alone software, which does not require the *PHOTOMOD* system installation. Also the program could be launched as a module of the *PHOTOMOD* system.

2. Installation of the module for neural processing of LIDAR data.

If *PHOTOMOD* has already been installed on the workstation, before installing the neural processing Module, please ensure that the *PHOTOMOD* and neural processing module version numbers and build numbers completely match.

To check *PHOTOMOD* version number and build number, perform the following:

1. Start *PHOTOMOD* (see the “General information” User Manual);
2. Choose **Help › About**. The **About** window opens:

Fig. 1. The About window

The **About** window displays a record of the following type: **Version: N.N.CCCC x64**, where **N.N** is the Version number, and **CCCC** is the Build number.

3. Close all running system modules:

   - Close the 2D program window (see “Work area interface” and “2D window” in the “General information” User Manual);

   - Close the *System Monitor* service module by right-clicking the icon of the appropriate item from the context menu in the OS notification bar (see the “System Monitor service module” in the “General information” User Manual).

The information on the neural processing of LIDAR data module version number and build number are contained in the module installer file name. The name of this file is the following: `PhNeuro_NN_[CCCC]_x64.deb` (or `PhNeuro_UAS_NN_[CCCC]_x64.deb`), where **NN** is the Version number, and **CCCC** is the Build number.

If the version numbers and build numbers of *PHOTOMOD* (or *PHOTOMOD UAS*) and the Neural processing of LIDAR data module do not match, update *PHOTOMOD* to the required version before installing the module.

3.1. PHOTOMOD Neuro installation

Make sure that *PHOTOMOD* (and/or *PHOTOMOD UAS*) is installed and configured. Make sure that all *PHOTOMOD* modules are closed.

To install *PHOTOMOD Neuro* perform the following:

1. Launch a **Terminal** window;

2. In **Terminal** window move to the folder containing `PhNeuro_NN_[CCCC]_x64.deb` or `PhNeuro_UAS_NN_[CCCC]_x64.deb` installation file;

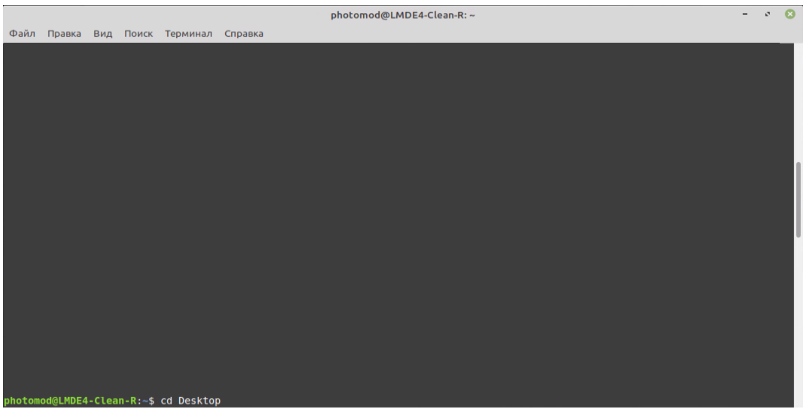

Fig. 2. The Terminal window

3. Type the following command in the **Terminal** prompt:

   ```
   sudo apt install ./PhNeuro_NN_[CCCC]_x64.deb
   ```

   or

   ```
   sudo apt install ./PhNeuro_UAS_NN_[CCCC]_x64.deb
   ```

   where **NN** is the Version number, and **CCCC** is the Build number.

   Press **Enter** to execute it.

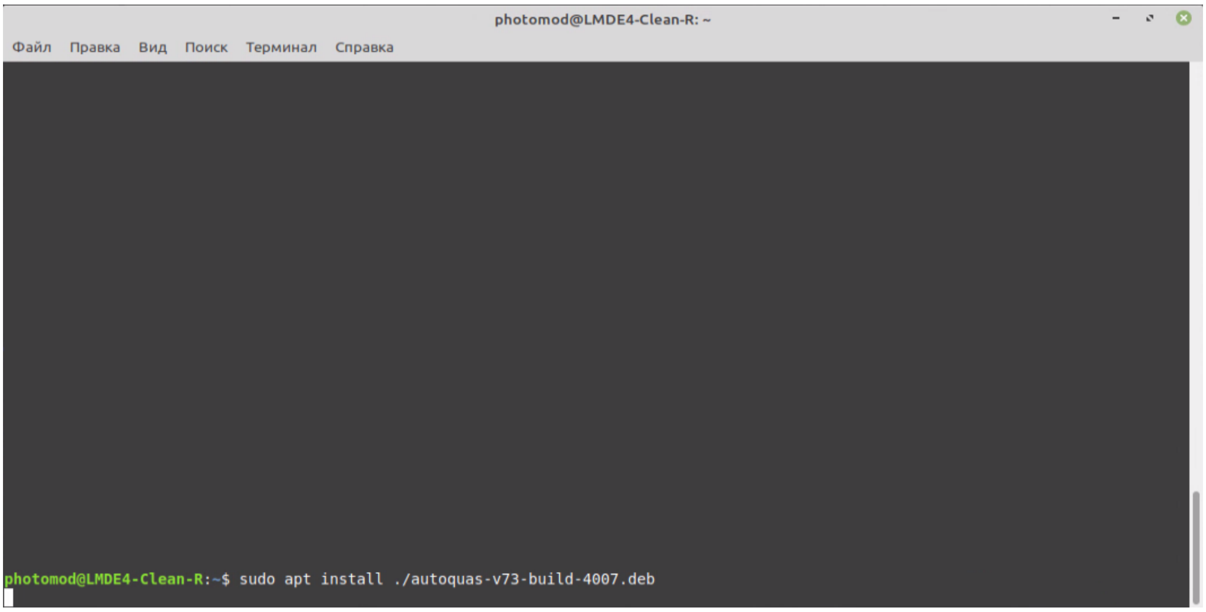

Fig. 3. The Terminal window

4. [optional] Confirm your action by entering your account password:

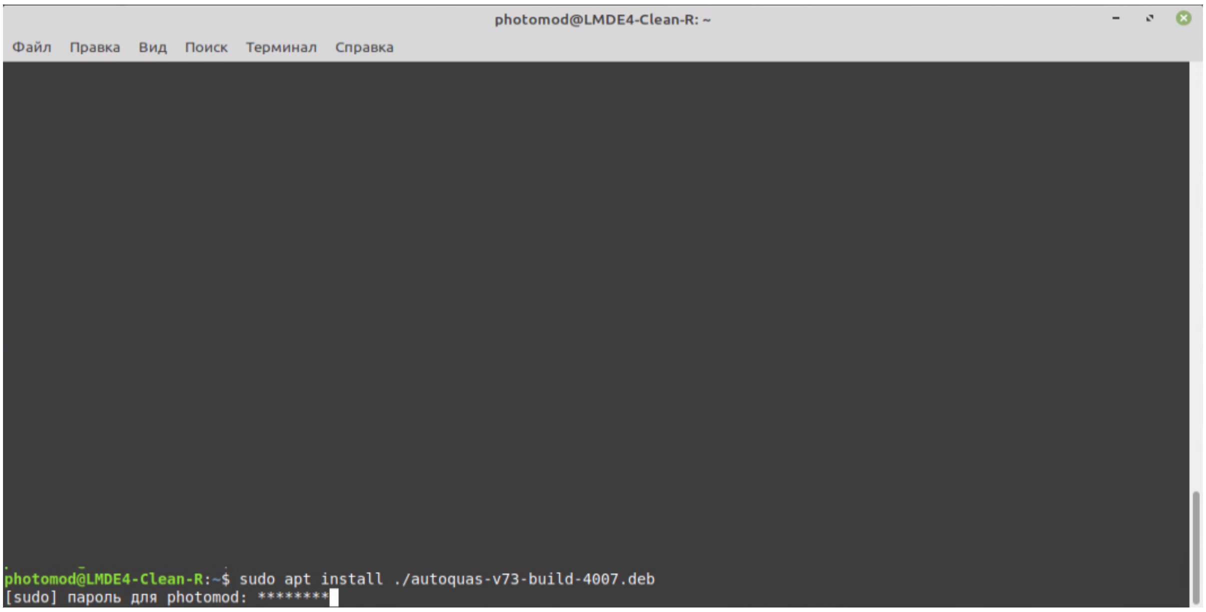

Fig. 4. The Terminal window

5. Wait until operation is completed;

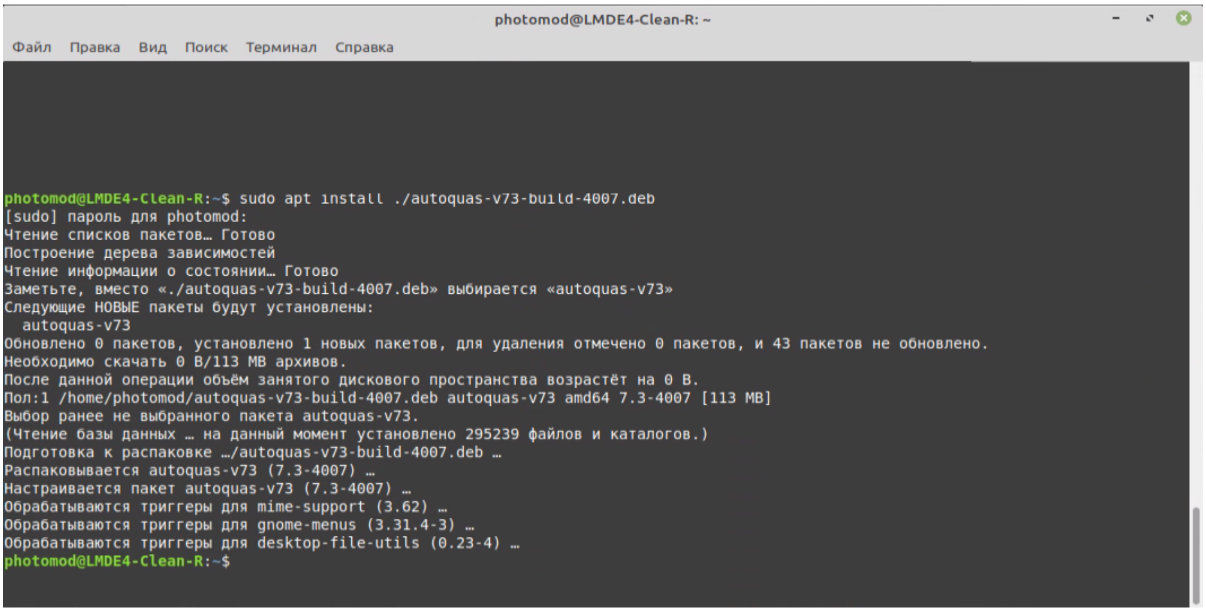

Fig. 5. The Terminal window

3.2. Software system start

To start *PHOTOMOD*, perform the following

1. Insert the unique *Sentinel HL* security hardlock key into the workstation USB port. Make sure that the key drivers have been installed on this workstation (see the "Sentinel security key" in the "General information" User Manual);

2. Choose **Start › Science › PHOTOMOD 8.0**.

   To start *PHOTOMOD UAS* choose **Start › Science › PHOTOMOD UAS 8.0**.

   - wait until the *System Monitor* service module starts. In the OS notification bar, the icon appears (see the "System Monitor service module" in the "General information" User Manual);

   - wait until the *PHOTOMOD* user interface opens:

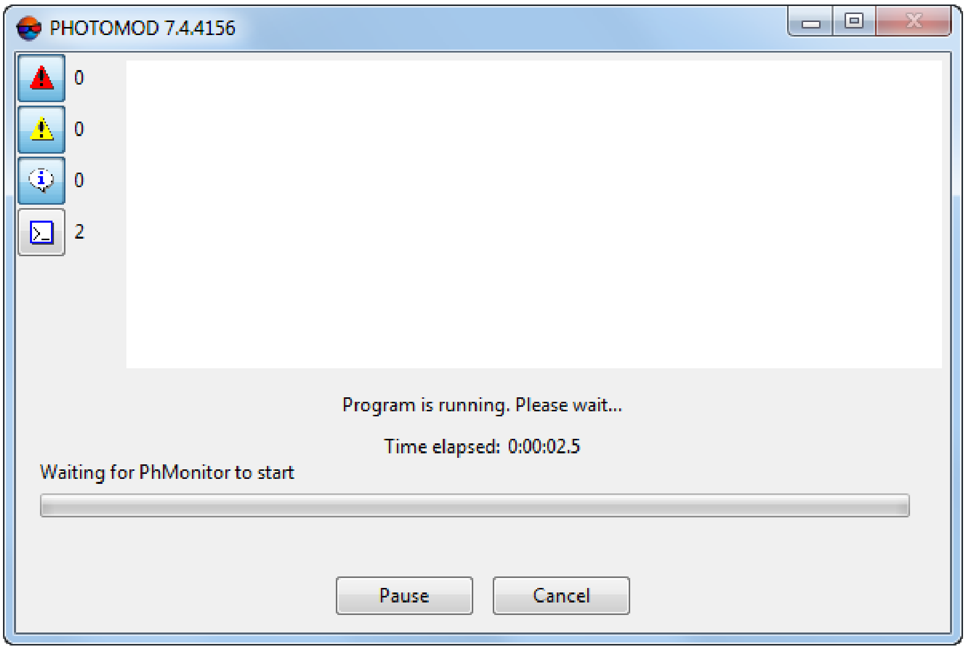

Fig. 6. PHOTOMOD loading progress window

To complete the user interface loading, perform one of the following:

- [optional] Click **w/o project** in the **Project management** window that opens

  Working with point clouds generally does not require loading any project, unless otherwise qualified (see the "LIDAR data processing" User Manual).

- [optional] Select the project that contains the required data and click **Open**.

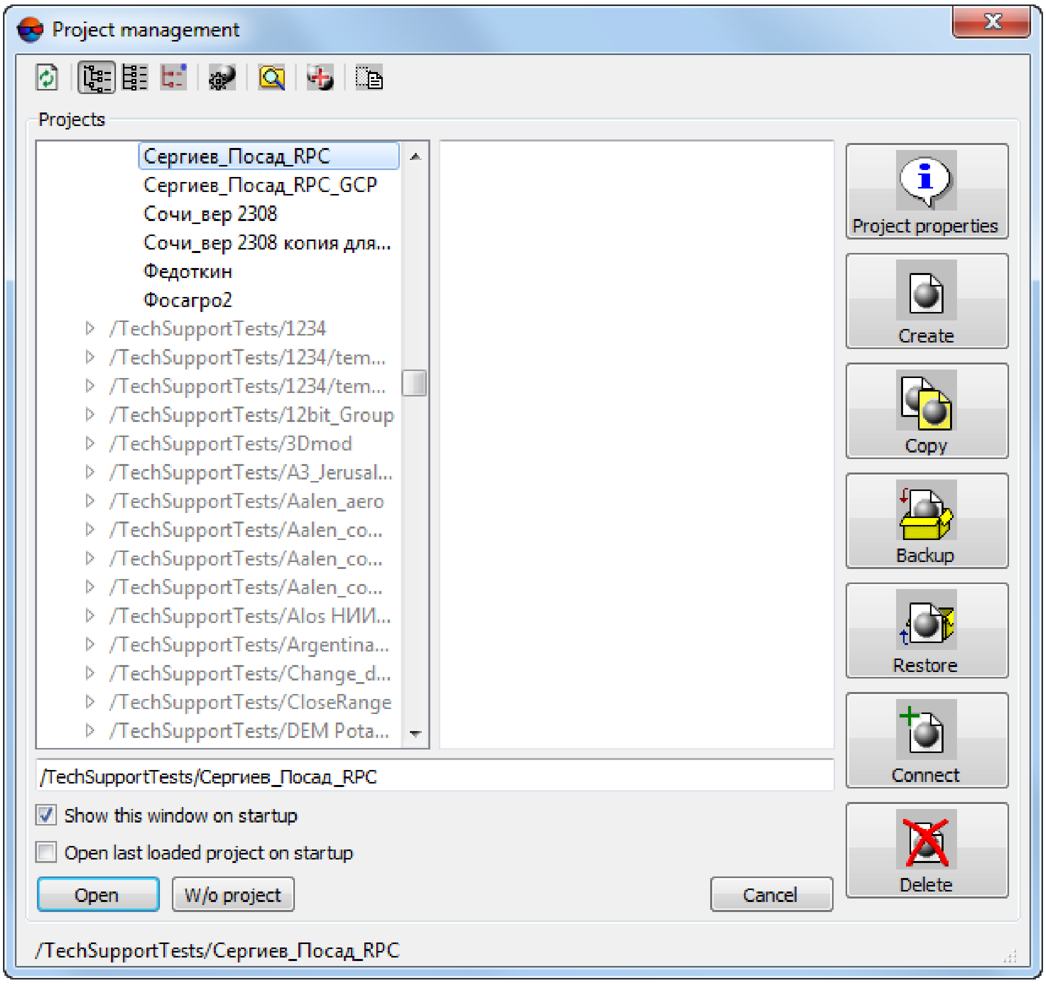

Fig. 7. The “Project management” window

- [optional] Start and configure the **Distributed processing control center**, if it is planned to use the distributed processing mode for work with large volumes of data (see the “Distributed processing” in the “General information” User Manual).

After installing the neural processing of LIDAR data module, an additional **Neuro classification methods** item will appear in the **Terrain › LAS** menu. This item provides an access to a submenu that contains the following tools:

Table 1. “Neuro classification methods” menu summary

Menu items	Function
Train point cloud classifier	Opens the **Train point cloud classifier** window
Classify point cloud	Opens the **Classify point cloud** window

4. Recommended data processing sequence

Recommended lidar point classification pattern using the neural processing of LIDAR data module includes the following steps:

1. Adding files that contain point clouds into the *PHOTOMOD* active profile resources:

   - Input data is the source point cloud, presented either as a single LAS file or divided into fragments (tiles) placed in the same folder in the file system;
   - Output data is a LAS file or a folder containing LAS files located in the *PHOTOMOD* active profile resources.

   The system does not provide for neural processing of files containing point clouds that are located directly in the file system folders (see “Data storing” and “Profiles” in the “General information” User Manual).

   Before starting data processing, load input point clouds into the resource system employed by *PHOTOMOD* (see “Connect virtual folder” in the “General information” User Manual).

2. [optional] Point cloud pre-filtering

   - Input data are LAS files located in the *PHOTOMOD* active profile resources;
   - Output data are LAS files after erroneous data filtering that are saved in the active profile resources.

   Filtering is the removal of point cloud errors before learning and classification. Filtering of rough errors (so-called outliers) to remove obviously erroneous points from the original cloud is strongly recommended. Such points are usually characterized by a significant distance from the main cluster and form small groups of isolated points.

   Point cloud filtering is carried out by standard *PHOTOMOD* tools and can be either manual (selecting and deleting groups of points) or automatic using special filters (see “Selecting point groups”, “Manual deleting point groups”, “LAS statistical outlier removal”, “LAS bilateral filter” and “LAS fusion” in the “LIDAR data processing” User Manual).

3. [optional] test classification of a point cloud fragment using a *preset* (see Appendix A):

   - Input data is a preset of data processing options, a fragment of source point cloud (optional, after prefiltering);
   - Output data is a point cloud fragment that contains additional information about the classes assigned to lidar points.

     The test classification of a relatively small (but quite representative) fragment of a point cloud has several goals. First, test classification allows users to assess the suitability of the preset for classifying specific data.

Second, a classified point cloud fragment (after required manual secondary processing by the user) can be used as a representative sample from a total data set during neural network learning (see below).

The sample is representative when the selected point cloud fragment displays a sufficient number of examples of all types of objects that are expected to be detected in the final classification process.

The sample should also comprehensively reflect the natural and anthropogenic elements of the landscape presented in the entire input data set.

If the default preset is used for classification, the system will search for all object classes provided by the system. Points that are not involved in the display of the set of searched objects (that are outside the user-specified classification threshold) are marked as **Unclassified**.

Point-class data already contained in the input cloud (if any) will be overwritten. The output point cloud is always saved as a new file. Input point cloud overwriting (using the same resource to save the output point cloud) is not provided.

Point clouds to be classified must contain information on lidar point colors.

Control of the obtained results is carried out using standard *PHOTOMOD* tools (see the “Load LAS” in the “LIDAR data processing” User Manual).

4. [optional] Representative sampling from the total data set. Data sampling is a required preparatory stage before training a neural network, which, along with the selected learning mode (see below), largely determines the results of this operation.

   Recommended data sampling methods and neural network learning modes directly depend on the nature of the studied data and the results of test classification:

   - [optional] A point cloud describes a landscape on the earth’s surface containing natural and anthropogenic objects that is quite similar to the reference data corresponding to the preset.

     The test classification based on the above preset delivered a satisfactory outcome, requiring minor adjustments and improving the recognition of certain types of objects.

     In this case, fine-tuning of the preset used during the test classification (see below) is recommended. The point cloud fragments obtained during the test classification and processed by the program should be used as a sample (after their required adjustment by the user).

     Debugging means manual classification (or reclassification) of groups of points, aimed at correcting errors made during test classification, both in the case of incorrect assignment of classes to groups of points and if the necessary objects were not identified at all (**Unclassified**).

Manual classification is carried out using standard *PHOTOMOD* tools (see “The “Load LAS” window”, “Point cloud boundaries”, “Selecting point groups” and “Manual deleting point groups” in the “LIDAR data processing” User Manual).

In most cases, to ensure reliable operation of search algorithms, the sample must contain a considerable amount (on the order of hundreds) of correctly selected objects belonging to each of the desired classes.

Hence, when forming a sample, it is recommended to use already partially classified data, which requires the selection of only part of the required types of objects (for which increased recognition accuracy is required).

Assigning classes to points that are not involved in displaying the desired objects is not mandatory.

- [optional] If the input point cloud displays landscapes that differ significantly from those previously studied and test classification attempts using existing presets yielded unsatisfactory results, then such point cloud processing may require the creation of a new preset generated by the user during learning with a clean slate.

  With this learning mode, existing presets are not used as input data, and the sample used can be created entirely manually by the user (this may be acceptable from the point of view of labor intensity and time, if during classification the user needs to select objects belonging to a limited list of classes, which, in turn, can be assigned to groups of points using the tools provided by the *PHOTOMOD*).

  If the preset was developed by the user completely from the ground up (and it is not a modification of a pre-installed set of data processing options and is based solely on the user’s own data sample, classified only manually), then the sample used to create it must necessarily include examples of objects of all possible classes, from among those that are supposed to be identified during the final classification.

  If such a preset is used for classification, the list of types of required objects will be determined exclusively by the data sample used during learning. Object classes that the user has not included in the sample will not be considered. Before learning, the program analyzes the provided sample and issues appropriate system messages about errors in cases when all (or some) of the object classes provided by the system are missing.

Note that in both cases described above, the created sample is to be further divided before learning into two independent units of comparable size, used by the learning algorithms as *training* and *test* data. Each of these units, in turn, must sufficiently comply with the conditions of representativeness described above in the previous subsection

The selection of a separate unit of *test* data from the sample is necessary because a pre-estimate of the quality of the learning performed is to be carried out. Based on this estimate, the user can interrupt the learning process in advance if acceptable preliminary results are achieved, thereby reducing time costs at this stage of data processing (see below).

Fragments of a point cloud that are used as *training* and *test* data are not to be identical to each other.

5. [optional] neural network learning (or fine-tuning) is a stage of the creation of a set of data processing options (so called *preset*):

   - Input data:

     - A representative sample from the total data set is two point cloud fragments that contain classified point groups created either solely through manual processing of LAS files or as a result of partial debugging of data obtained during test classification (the latter is recommended in most cases);

     - [optional] One of the available presets, except in cases of irregular data set processing. If the input data for creating the sample were the test classification results, it is recommended to select the preset used during the test classification.

   - A learning result is a new (user) preset available in the program interface after performing the learning operation, generated specifically for processing a specific data set. The user preset is saved as a separate file in the file system, in the *PHOTOMOD PHOTOMOD8.VAR* setting folder.

     If the user creates a large number of presets, it is necessary to take into account that files containing sets of data processing options can take up a significant amount of space on the hard drive on which the *PHOTOMOD* settings folder is located.

     A custom set of data processing parameters can later be used both during fine-tuning and during classification, when processing other point clouds that describe landscapes with similar characteristics.

     It must be taken into account that the *distributed data processing mode* is not provided for the learning process. Accordingly, to ensure sufficient system performance, it is recommended to use a point cloud fragment of a relatively moderate size as a sample.

6. Final point cloud classification:

   - Input data:

     - A custom preset, either pre-installed or created during learning (see above);

     - An input point cloud (or pre-filtered point cloud, see above).

   - The final output data is the output point cloud, which contains additional information on the classes assigned to lidar points.

In the case of processing significant amounts of data, in order to increase system performance, use the *distributed data processing mode*.

5. Point cloud classifier learning

The system allows the user to create a new preset through point cloud classifier learning based on a user-prepared sample from the total data set.

The classifier training algorithm is as follows:

1. The system has *training* and *test* data sets available, for example, fragments of a large data unit that needs to be classified.

   The *training* and *test* fragments are correctly classified in advance (for example, manually), they are not identical, but describe similar territories and include the same sets of classes;

2. The system (using *training* data as an example) determines the characteristic features of points describing objects belonging to a certain class;

3. Taking into account the information received, multiple sessions of classification of the *test* data set are performed. For this purpose, the system does not rewrite the classes in the *test* set, but repeatedly compares the obtained results with the classification originally available in it.

   Step by step, changing the weight of the analyzed characteristics of the points with each iteration, the program, in the ideal case, achieves maximum coincidence of the classification results with the *test* set (or the iteration cycle is interrupted by the user at a certain point);

4. The most optimal set of data processing parameters obtained in this way is saved as a training result for further classification of the main data array

For this, perform the following:

1. Select two pre-processed point cloud fragments that meet the requirements described above;

2. Choose **Terrain › LAS › Neuro classification methods › Train point cloud classifier**. The **LAS classifier training** window opens:

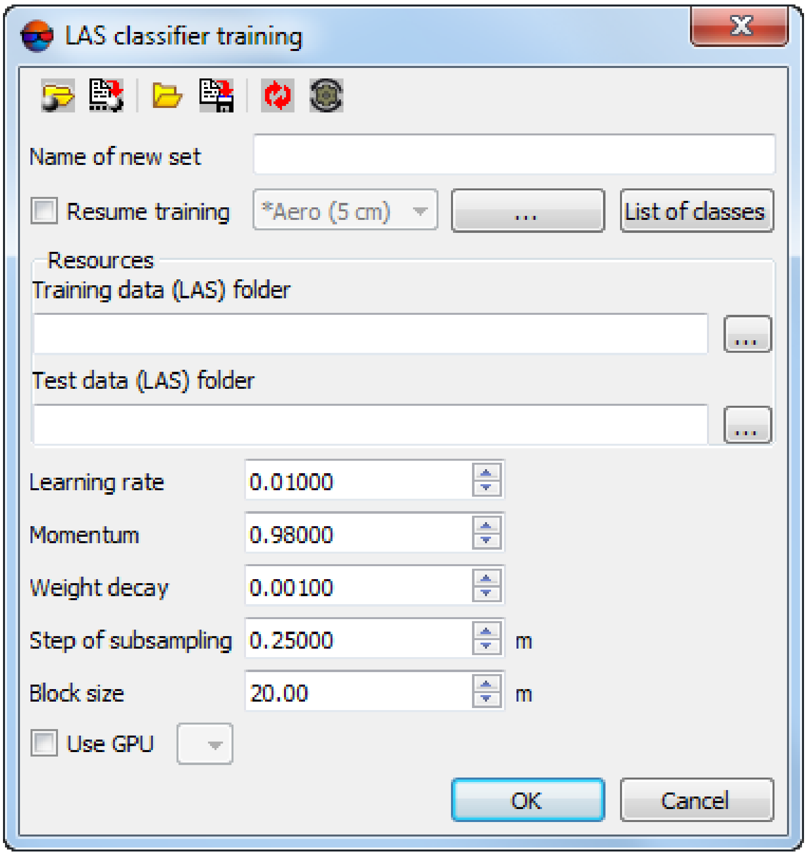

Fig. 8. The “LAS classifier training” window

The **Train point cloud classifier** window has the following interface elements:

* A toolbar in the upper part is to save/load data processing options. It includes the following buttons:
  * – to load options previously saved in the active profile resources;
  * – to save all option settings in the `*.x-ini` file in the active profile resources;
  * – to load option settings from the `*.x-ini` file in the file system folder;
  * – to save option settings in the `*.x-ini` file in the file system folder;
  * – to restore the settings of the previous data processing session;
  * – to restore default settings.
* The **Resources** subsection and the subsection to set data processing options.

3. Choose used input **resources** in the appropriate subsection:

- **Training data (LAS) folder** – click ... to choose a catalog containing the prepared LAS point cloud fragment in the active profile resources;
- **Test data (LAS) folder** – click ... to choose a catalog containing the prepared LAS point cloud fragment in the active profile resources.

Fragments of a point cloud that are used as *training* and *test* data are not to be identical to each other

4. In the **name of new set** field, set any name of the user preset to be created;
5. To **resume training** of one of the existing presets (without creating a custom preset "from scratch"), set the appropriate checkbox. Select the existing *preset* from the drop-down list.

   To handle the available presets, click the ... button. The **Configurations of parameters of the neural network** window opens that allows the user to **delete** or **rename** the available presets:

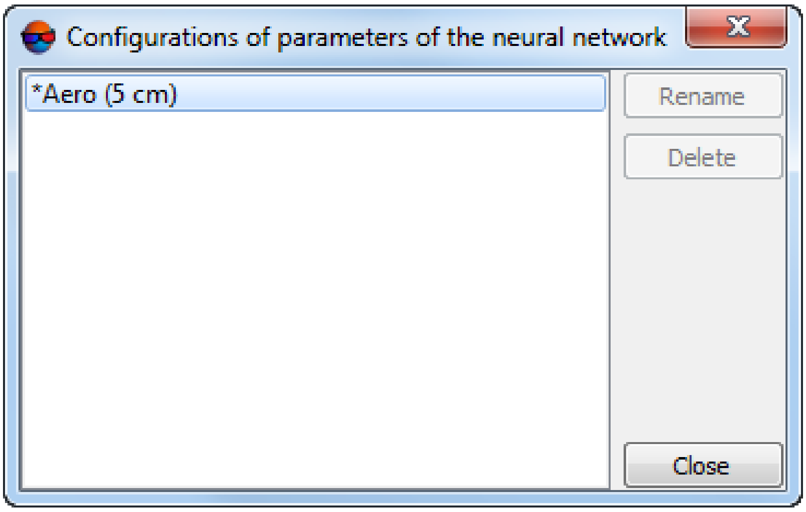

Fig. 9. The "Configurations of parameters of the neural network" window

6. [optional] To manually configure the **list of classes** considered during learning, click the appropriate button. The **list of classes** window opens. All *ASPRS*-provided classes are included in this list, as well as three additional classes the system uses, i.e., **cars**, **road furniture** and **wall**.

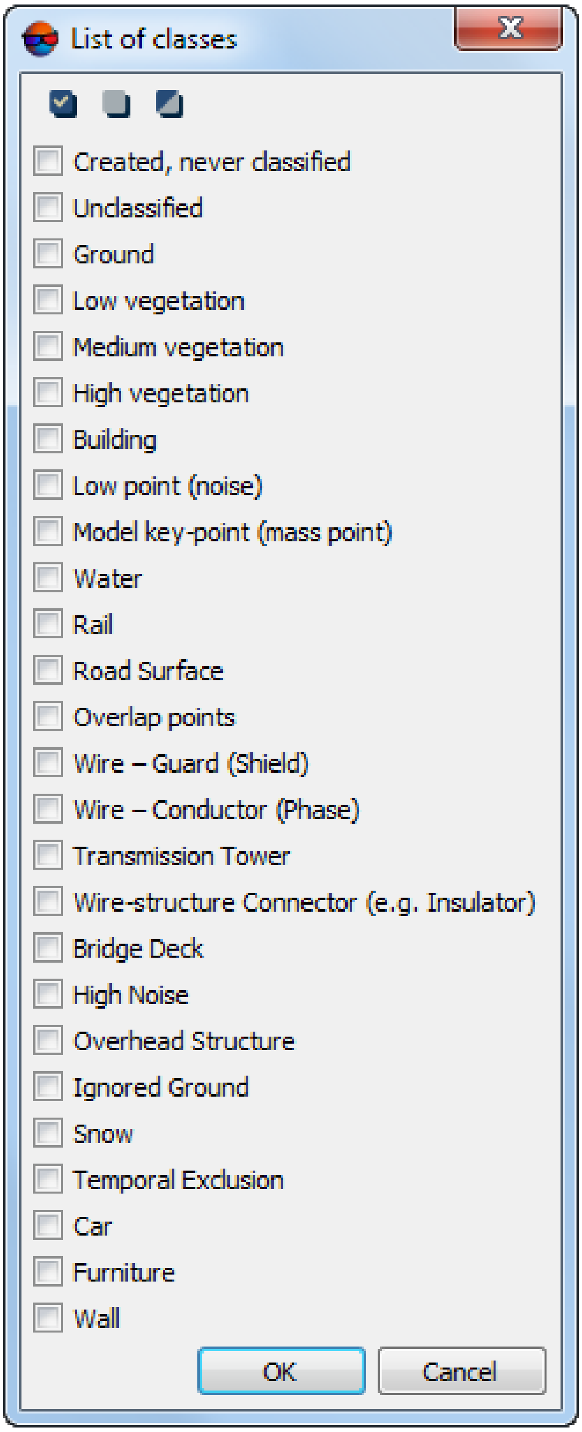

Fig. 10. The “List of classes” window

To exclude certain classes from the preset creation process, clear the appropriate checkboxes. If the **resume training** checkbox was preselected, the point classes that are already present in the selected preset are highlighted by bold font.

The following is strongly encouraged:

- *Training* and *test* data must contain the same set of lidar point classes;
- All classes selected in the **list of classes** must belong to the preset mentioned above.

At least two classes are to be involved in the preset creation.

For group selection of point classes, the following buttons are provided in the **list of classes** window:

- – to select all classes;
- – to deselect all classes;
- – to invert the selection of classes.

7. Configure the following options:

- The **Learning rate** is the initial coefficient of weight increment for one iteration of data processing, gradually decreasing during the task execution process with each new learning cycle. Increasing the weight increment factor can increase system performance while reducing the quality of data processing;

  In most cases it is not recommended to change the default value of this parameter.

  The learning process consists of global cycles, including a multitude of iterations. The user can watch the process of learning in the special progress window (see below). After the end of each learning cycle, an appropriate system message is displayed in the progress window containing a brief report on the results of the completed cycle.

  The user is able to either interrupt the learning ahead of schedule (upon achieving acceptable results from their point of view) or wait for the automatic completion of the learning process by the program itself (the latter may require significant time costs).

- **Additional value to the descent gradient** – the error influence coefficient in previous iterations. It is recommended to increase or decrease the error influence coefficient in direct dependence on the degree of heterogeneity of objects displayed by the point cloud under study

  In most cases it is not recommended to change the default value.

- **Libra loss (fine)** – the coefficient of regularization of the artificial neural network parameters;

  In most cases it is not recommended to change the default value.

- Input point cloud **step of thinning** before processing, in meters. In most cases, to ensure an optimal balance between the quality of learning and system performance, it is recommended to set the **step of thinning** to approximately five times the resolution of the input point cloud (average distance between lidar points);

- The **block radius**, in meters, which determines the size of the studied point cloud fragment in the vicinity of an individual lidar point. The recommended **block radius** should be comparable to the linear dimensions of most objects requiring classification. A significant increase of the **block radius** can negatively affect system performance.

  If the *preset* created during the current learning session will be subsequently used to classify the point cloud, then when setting the classification parameters, it is strongly recommended to set the same (or comparable) **block radius**.

  Neural processing of large arrays of lidar data places high demands on the workstation RAM. In cases of relevant system messages about insufficient RAM, it is recommended to reduce the **block radius**.

8. To increase the system performance due by using the video adapter resources, set the **Use GPU** checkbox and choose the desired device from the drop-down list;

   Monitoring GPU usage activity is possible using various free software.

   The system allows to **use GPU** during the neural processing only on PCs with *NVidia* video adapters.

9. Click OK. The user can check the operation progress in the progress bar that opens:

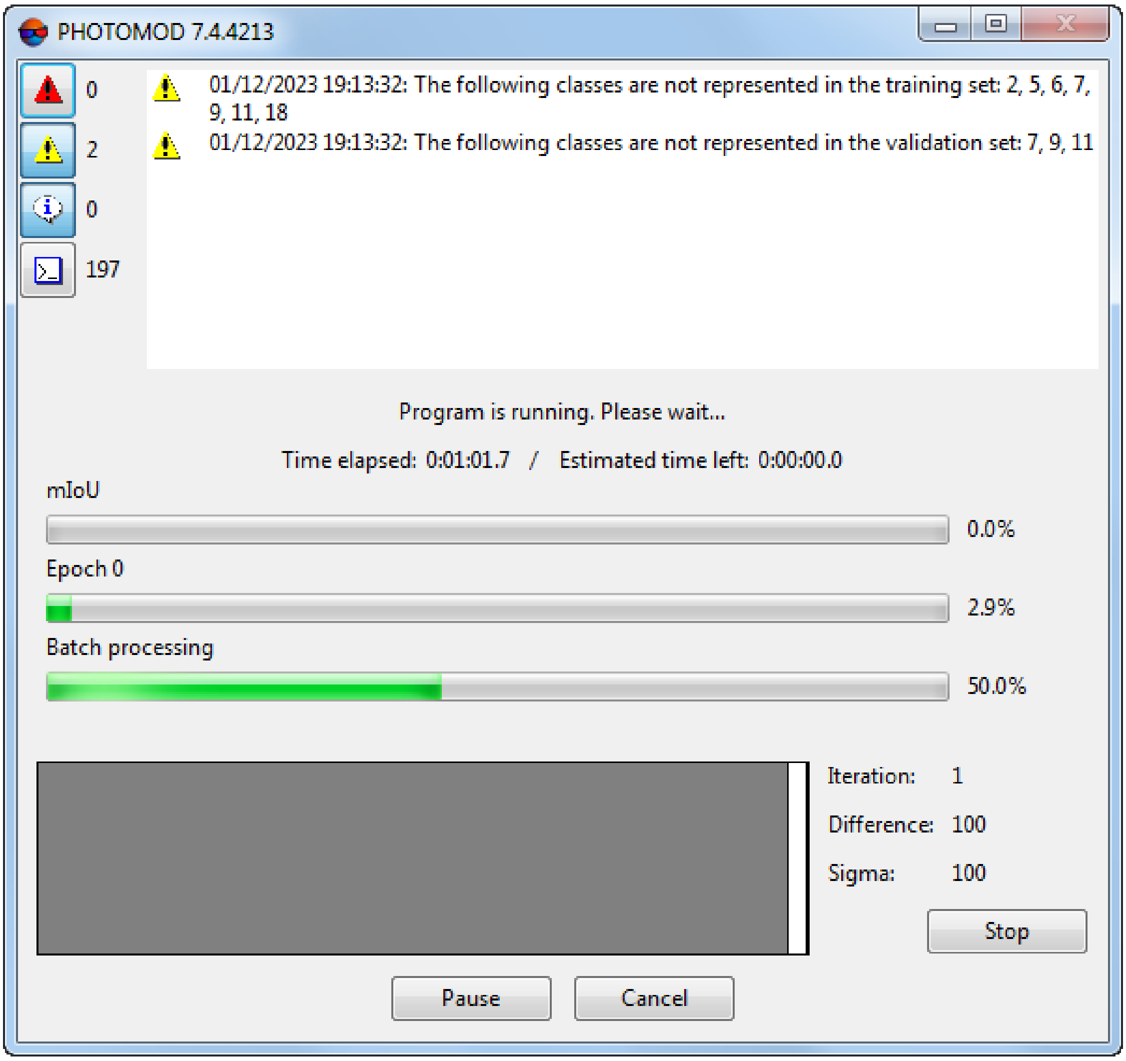

Fig. 11. The progress window

The progress window has the following interface elements:

- An area to display system messages, in particular:
  - [optional] Warnings about the lack of one or more classes of objects provided by the system in the training data;
  - [optional] Warnings about the lack of one or more classes of objects provided by the system in the reference data;
  - [optional] A system message that the user provides incorrect input data (in this case, the learning process will be cancelled);
  - System messages informing the user about the completion of the learning cycle and containing a brief report on the results of the completed cycle.

- An area displaying the progress of actual operations;
- An area displaying the number and results of completed learning cycles in the form of a diagram and a set of current parameters (is displayed after the input data preload is complete).

A *training* data set is used for learning. The *test* data set is used to constantly validate the quality of object recognition by the created set.

The dynamics of assessing the quality of neural network learning from the system side are reflected in the changing values of the **Difference** and **Sigma** parameters. Stabilization of these values at a certain level, after an initial decrease during the first learning cycles, may indicate that the program may have achieved optimal processing parameters for the provided sample data.

The **Stop** button interrupts the learning process to capture current results. If the learning process is successfully completed (in automatic mode or as a result of user actions), a corresponding system message is displayed in the progress window:

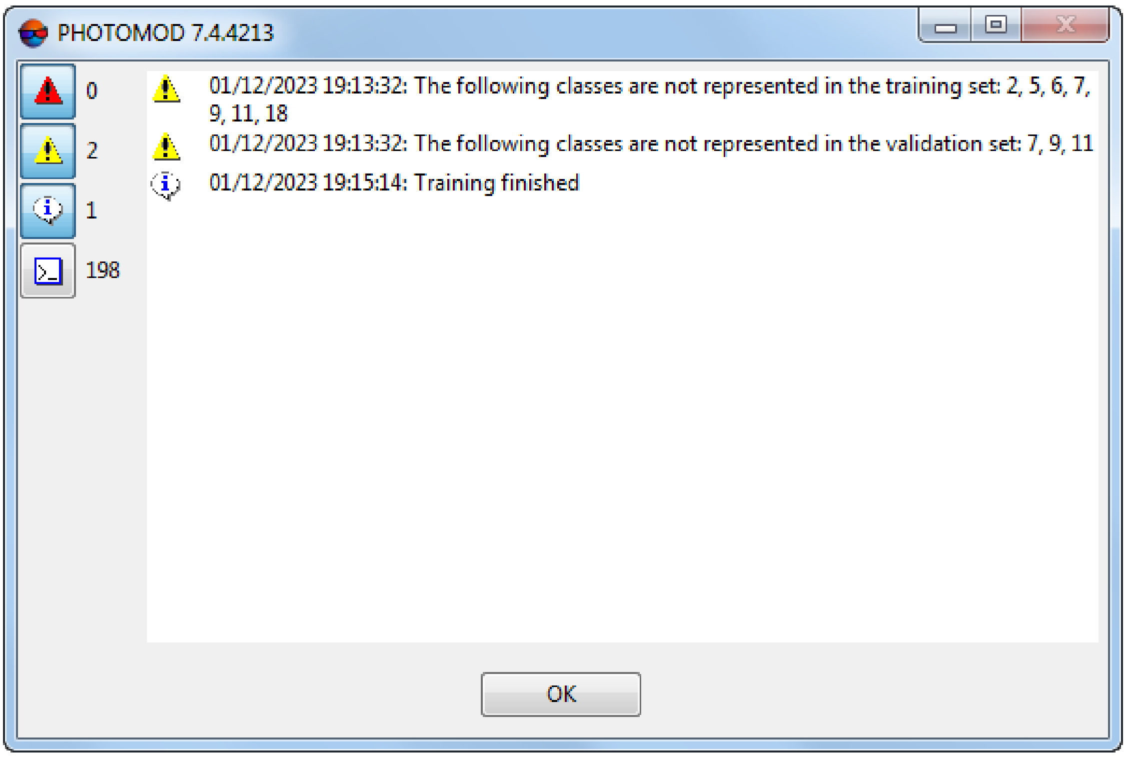

Fig. 12. The progress window

A new set of data processing options created as a result of learning is displayed in the appropriate sections of the **Train point cloud classifier** and **Classify point cloud** windows.

6. Point cloud classification

The system provides for lidar point classification by neural processing.

For this, perform the following:

1. Choose **Terrain › LAS › Neuro classification methods › Classify point cloud**. The **LAS classification** window opens:

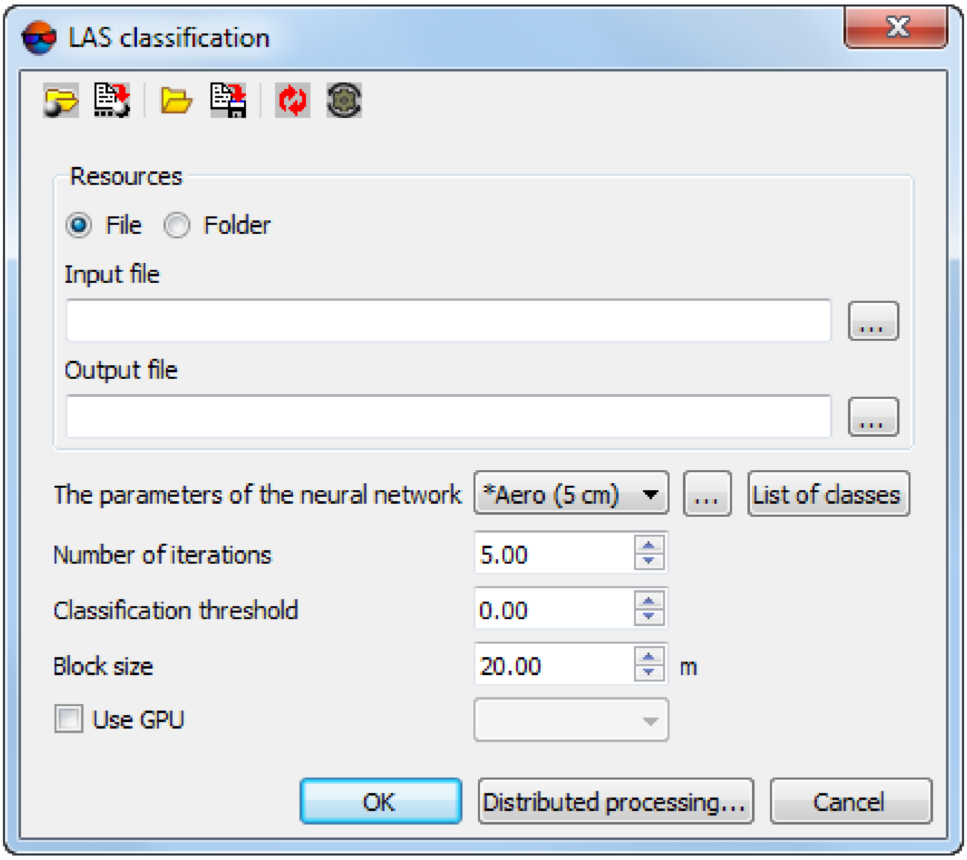

Fig. 13. The “LAS classification” window

The **LAS classification** window has the following interface elements:

- A toolbar in the upper part is to save/load data processing options. It includes the following buttons:
  -  – to load options previously saved in the active profile resources;
  -  – to save all option settings in the `*.x-ini` file in the active profile resources;
  -  – to load option settings from the `*.x-ini` file in the file system folder;
  -  – to save option settings in the `*.x-ini` file in the file system folder;

- ○ – to restore the settings of the previous data processing session;
- ○ – to restore default settings.

• The **Resources** subsection and the subsection to set data processing options.

2. Select the input **resources** to use in the appropriate section:

   • [optional] **LAS file** – click ... to select the **input LAS file** in the active profile resources;

   To process a point cloud saved as a single file.

   • [optional] **LAS folder** – click ... to select the **input point cloud (LAS) folder** containing fragments (tiles) in the active profile resources

   To process a point cloud divided into fragments (tiles).

3. In the **Output point cloud (LAS) folder** field (or in the **Output LAS file** field), enter the path to save LAS files or click , to select the name and path in the active profile resources;

4. Configure the following parameters:

   • In **the parameters of the neural network** section, select one of available data processing options from the list. To manage the available array of presets, click the ... button. The **Configurations of parameters of the neural network** window opens, which allows the user to **delete** or **rename** available presets

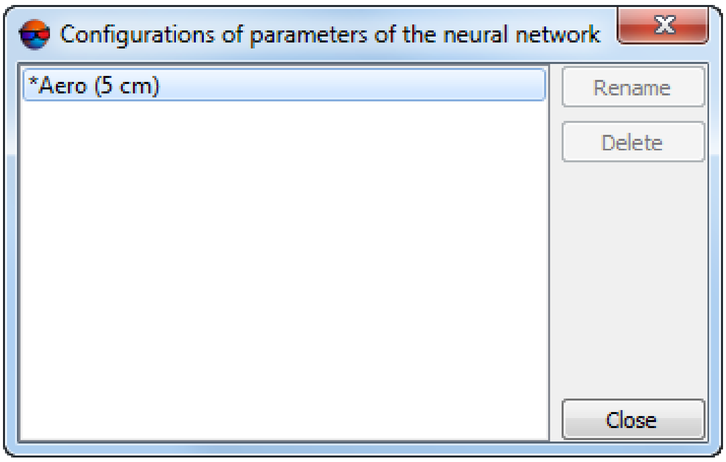

Fig. 14. The “Configurations of parameters of the neural network” window

- Set the **number of iterations** (classification attempts) for an individual point;
- Set the lidar point **classification threshold** (between 0 and 1);

  Points that are not involved in displaying the set of searched objects (that are outside the specified classification threshold specified by the user) are assigned the **Unclassified** class.

- The **block radius**, in meters, which determines the size of the studied point cloud fragment in the vicinity of an individual lidar point. The recommended **block radius** should be comparable to the linear dimensions of most objects requiring classification. A significant increase of the **block radius** can negatively affect system performance.

  If a custom set of data processing parameters is selected in the neural network parameters section, it is recommended to enter the same value for the **block radius** as that specified during the learning process, as a result of which the selected *preset* was created.

  Neural processing of large arrays of lidar data places high demands on the workstation RAM. In cases of relevant system messages about insufficient RAM, it is recommended to reduce the **block radius**.

  If the distributed processing mode is used, it is also recommended to reduce the maximum number of MT tasks

5. [optional] To configure manually the **list of classes** considered during classification, click the appropriate button. The **list of classes** window opens. All classes of the selected preset are included in this list:

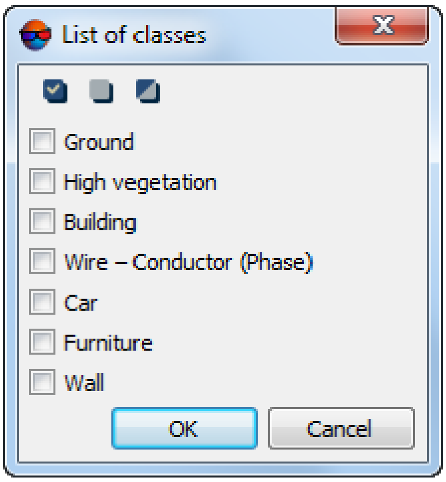

Fig. 15. The “List of classes” window

To exclude certain classes from the classification process, clear the appropriate checkboxes.

At least one class should be selected.

For group selection of point classes, the following buttons are provided in the **list of classes** window:

- – to select all classes;
- – to deselect all classes;
- – to invert the selection of classes.

6. To increase the system performance due by using the video adapter resources, set the **Use GPU** checkbox and choose the desired device from the drop-down list;

Monitoring GPU usage activity is possible using various free software.

During distributed processing, video adapters are to be selected in the **Monitor for Distributed Processing** window, individually for each computer used as a distributed processing client (see “Computers” in the “General information” User Manual. However, video adapters’ resources will be used for distributed computations only if the **Use GPU** checkbox is set in the current window.

If the computer used to configure the operation options (*server*) is also used as one of the *clients*, the selection of the video adapter used by it is also carried out through the **Monitor for Distributed Processing**, regardless of which device was selected in the drop-down list in the current window (this choice is taken into account only in the case of standard data processing mode).

The system allows to **use GPU** during the neural processing only on PCs with *NVidia* video adapters.

7. Click OK.

To classify a LAS point cloud using distributed processing, perform the following:

1. Configure and start the distributed processing server/client (see “Distributed processing” in the “General information” User Manual)
2. Click the **Distributed processing** button.

After successful completion of the operation, an appropriate information message is issued:

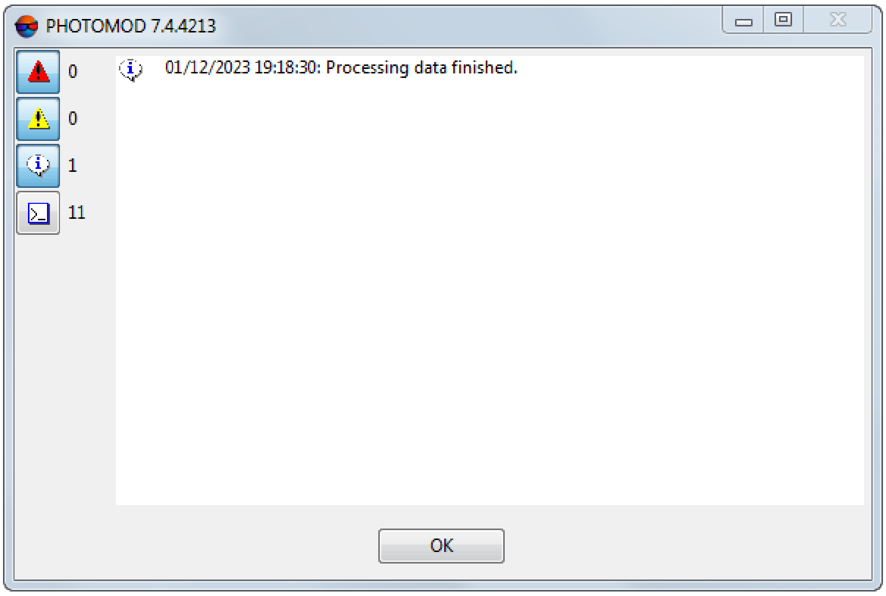

Fig. 16. The information message

The obtained results are checked using standard *PHOTOMOD* tools (see “The “Load LAS” window” in the “LIDAR data processing” User Manual).

Appendix A. The “Aero (5 cm)” preset

The standard **Aero (5 cm)** (GSD 5 cm) preset is a preset of data processing options and a relevant set of reference data. The reference data is a point cloud created by the photogrammetric method, describing with a high degree of detail the landscape characteristic of a small European town and including objects belonging to all classes provided by the system:

- Earth surface, i.e. sidewalks, walkways, alleys;
- High vegetation, i.e. trees, shrubs, hedges;
- Buildings, i.e. fences, roadside barriers, walls, bridges, parking lots, railroads;
- Cars;
- Furniture, i.e. benches, utility poles, street lights.

Using the **Aero (5 cm)** preset (unmodified) directly to classify point clouds describing areas of the earth’s surface on the territory of the Russian Federation may, in some cases, be impractical due to the fact that the nature and shape of buildings and structures presented in the reference example are not always similar to typical buildings characteristic of most settlements in Russia.

However, this preset includes all provided classes of objects and is strongly recommended to be taken into account when creating your own set of data processing parameters (see Section 5).

Table A.1. LIDAR points classification provided by “Aero (5 cm)” preset

Classification Value	Meaning
1	Created, never classified
2	Unclassified
3	Ground
4	High Vegetation
5	Building
10	Furniture
12	Cars

☞ **Cars** and **furniture** LIDAR point classes are used by the program but are not part of the standard *ASPRS* classification. The program employs the “user definable” classes from the standard *ASPRS* classification, treating them as cars and road accessories within the system.