User Guide

TIDOP

September 15, 2021

1 Overview

This software aims at classifying automatically points clouds thanks to a small ground-truth subset given by the user. It takes a partly classified point cloud as input.

Classification can be done by manually giving features and their individual weights (positive or negative) towards each class. The second option uses these features and the training algorithm from CGAL library to automatically compute the weights. Lastly, a deep learning algorithm is proposed to train and classify based on original point cloud features.

2 Pre-processing data

First, you will need to process your data in .ply format, with a "label" int property indicating the class of the point, beginning from 0. Unclassified points should be assigned value -1.

```
ply
format ascii 1.0
comment Created by CloudCompare v2.11.3 (Anoia)
comment Created 15/09/2021 10:34
obj_info Generated by CloudCompare!
element vertex 189219
property float x
property float y
property float z
property uchar red
property uchar green
property uchar blue
property int label
end_header
50.0638 22.5993 604.312 51 51 53 -1
50.0648 22.6064 604.306 45 45 45 -1
50.0628 22.6134 604.3 47 47 49 -1
50.0648 22.6173 604.306 44 42 45 -1
50.0638 22.6173 604.311 43 41 44 -1
50.1118 22.6215 604.292 41 40 45 -1
50.1058 22.6254 604.303 43 41 46 7
50.1038 22.6264 604.306 43 41 46 7
50.1068 22.6244 604.299 41 40 45 7
50.0958 22.6403 604.295 45 43 54 0
50.1088 22.6315 604.293 41 40 46 0
50.1068 22.6403 604.291 41 41 51 0
50.1028 22.6335 604.3 41 40 46 5
50.1018 22.6415 604.302 46 43 50 5
50.1018 22.6344 604.308 46 43 50 5
50.1088 22.6335 604.297 41 40 46 5
50.0678 22.6034 604.318 46 44 47 5
50.0638 22.6093 604.313 46 44 47 5
50.0628 22.6115 604.32 52 47 51 7
50.0658 22.6115 604.316 46 44 47 7
50.0698 22.6024 604.323 52 47 51 7
```

Figure 1: Base windows

You can use CloudCompare to assign colors to points depending on their class. After saving it in .ply file, use the python program "labeler" to select the file and choose the color you gave to unclassified points.

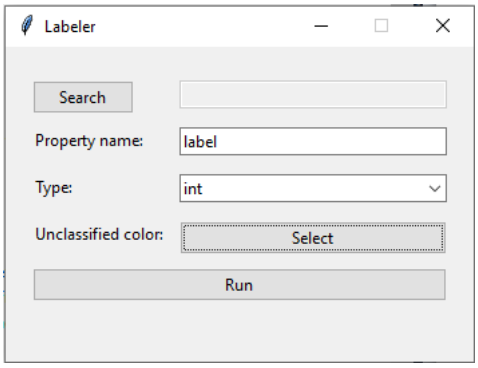

Figure 2: Base windows

3 Application

The application features the following :

- **Add label** : Allows to add label. Color should be changed and distinct as RGB values are what will represent point class at the end.
- **Add feature** : Allows to add features from the ones proposed by CGAL (elevation, distance to plane...) and assign it a weight representing its importance. Only used for manual classification.
- **Add effect** : Allows to add the effect of a feature towards a label, which can be neutral, penalizing or favoring. For instance, columns often have a high "verticality" score, so verticality would be a favoring feature towards the column class. Only used for manual classification.
- **Run classification** : After having added labels, features and effects, classifies unclassified parts of the point cloud. Only used for manual classification.
- **Train** : After having added labels, launches training to automatically determine weights and effects. Only used for automatic classification.
- **Tools menu and Save button** : Unused at the moment.

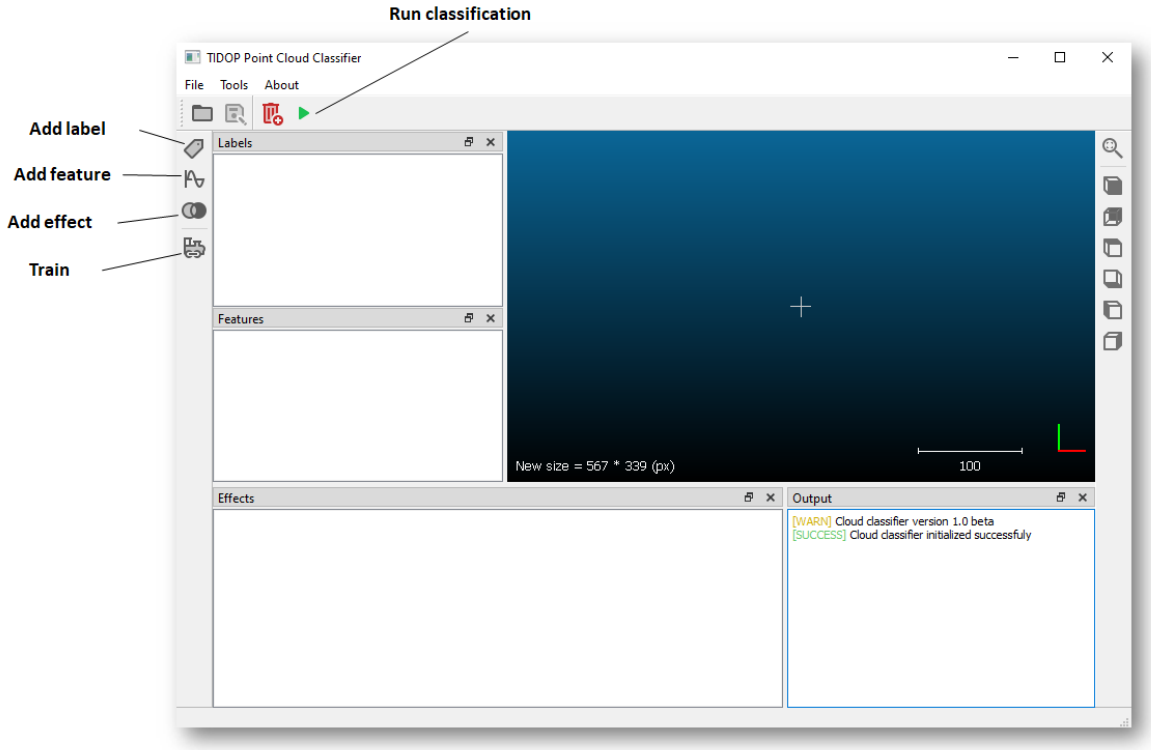

Figure 3: Base windows

4 Manual Classification

The classification section of the program is based on CGAL, more information can be found on CGAL official documentation. To classify a point cloud, add labels and assign different colors to the output. Add features which seem relevant to your use case. Then add the effect of each feature towards each class.

You can then push the "Run classification" button to launch the classification. A .ply file will be created where the original file was taken from with the original file name plus an "-classified.ply" appendix.

5 Automatic Classification

The classification section of the program is based on CGAL, more information can be found on CGAL official documentation. To classify a point cloud, add labels, assign colors and click on train to show the following output :

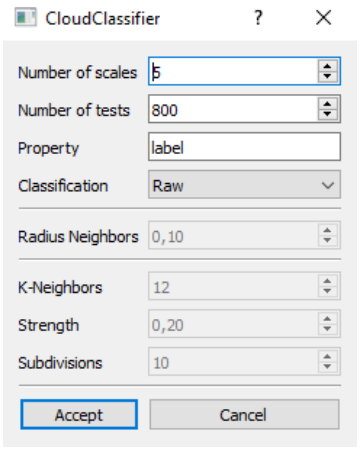

Figure 4: training windows

- **number of scales** : number of different scales for neighbor computations. The more the better and the slower (exponentially).
- **number of test** : number of iterations to find weights and effects. The more the better and slower.
- **Property** : which property defines the class of each point
- **Classification type** : type of classification to use, ranging from the fastest but crude "Raw" to the slowest but with the best results (Graph-Cut).

A progress bar will appear. As a reference, 200 000 points in training with 8 classes takes a few minutes to compute.

Once training is finished, a .ply file will be created where the original file was taken from with the original file name plus an "-classified.ply" appendix. An additional xml file will save the features, weights and effects computed.

6 Deep learning Classification

Not implemented

7 Troubleshooting

Restart the application and try again