JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 27 No. 2 (2019), pp. 7-16

Classification of Objects in a Point Cloud using Neural Networks

Marcin Daszuta, Ewa Napieralska-Juszczak[1]

[1] *LSEE, Univ. Artois*
Technoparc Futura
62400 Bethune, France
ewa.napieralskajuszczak@univ-artois.fr

Abstract. *3-dimensional scans captured in shape of point clouds are widely used in many different areas. Every such area use different kinds of sensors to ac-quire point clouds and do the analysis of the data but each of those needs some preanalysis to be done. One of the most important is segmentation and classification of points into types of objects. Such information considerably widens possibilities of usage for further purposes. There are many classifiers and many features based on which labeling can be done. In this paper few most commonly used approaches were chosen to check the influence of neighboring points acquisition on classification process. Results proof signif-icant relation between those two steps of point cloud analysis. Visualization of analyzed point cloud also shown that precision of predictions not always comes with better visibility of certain types of objects. Additionally, color-less analysis of geometrical features seems to be promising way for further research.*
Keywords: *Point cloud classification, Voxel Partitioning, KNNR, Random Forest.*

1. Introduction

Analysis of 3-dimensional scans are currently quite widely discussed in many different areas, such as robotics, augmented reality or remote sensing. Each of these areas can use sensors to get data from environment and use it. However, to allow it scan has to be processed to the level which allows getting certain types of information. For example, what type of object is visible. To do this, parts of 3-D scan should be segmented and labeled. Such scan is usually acquired by Light Detection And Ranging (LiDAR), Red Green Blue-Depth (RGB-D) cameras and Synthetic Aperture Radar (SAR) systems [1][2]. Each of those methods generate scan in shape of point cloud with different additional data, for instance color.

However, not every type of sensor allows to capture color or some additional information, which would allow to augment data about scanned environment. That is why basically, safe data is only position of each point captured by scan-ner in 3-dimensional space. Focusing on position of points and entropy of neigh-boring points it is possible to conclude certain characteristics like geometrical features. These are used to train machine learning algorithms which with certain level of precision are able to decide if point is part of known object type [3][4]. Good example could be points labeled as chair or table in point cloud.

Through such semantic segmentation grouping of points can be achieved. However, accuracy of this process depends on many different aspects. For instance, method of choosing neighborhood for each point can differentiate values of features, as those are acquired through covariance matrix [5][6][7]. Another issue is choice of the best classifier for this kind of problem. This paper focus on showing relation between neighborhood selection per point and semantic classification accuracy.

2. Related work

There are few commonly known methods of neighborhood selection. Some are based on division of cloud point space into patches for example, voxels. where other focus on processing points separately. Each approach has some drawbacks and assets [8].

2.1. Voxel partitioning

Voxels can be achieved by partitioning cloud point by few different methods, like k-d trees or octree. Though it is mainly used to narrow down number of points that need further analysis, since in such case it is possible to focus only on so called centroids. Mean point calculated as center of voxel. Such methods greatly reduce amount of calculations needed to finalize segmentation, with cost of accuracy. This kind of division is also vulnerable on issue with number of points considered as neighborhood [9].

2.2. K-nearest neighbors (KNN)

Another idea is to process each point with consideration of K-nearest points [10]. In this case, only constant is number of points treated as neighbors. This ap-proach secures situation when in the closest surrounding it is not possible to gather enough points for geometrical features acquisition. However, since there is rule to acquire certain number of points is happens that neighbors are very far from processed point. It also causes certain inaccuracy during feature extraction. Such approach is more accurate than voxel division, though it still can cause feature fluctuations with the reason in high irregularity of neighborhood range [4]. Good example is Fig. 1. When we choose K as 10 it is visible that points very far away from processed point (red dot) will also have influence in further calculations.

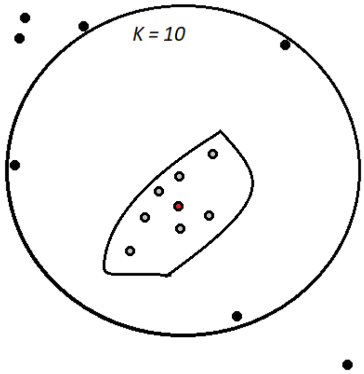

Figure 1: Example of K-nearest neighbors choice with K=10

2.3. K-nearest neighbors in range (KNNR)

To avoid such fluctuation, it is possible to add one more condition, bounding searching area to certain range. In this case if chosen K of neighbors is not possible to consider in chosen range searching shall be terminated with number of neighbors already acquired. Additionally, if there is more points than K in chosen range, then acquisition will be finished after reaching K points. Therefore, it is possible to assign surrounding more regularly but per every point individually. Unfortunately, it also comes with increased processing cost [4].

2.4. Geometric Features

As it was mentioned earlier, there are certain geometric features considered during point cloud analysis. The most commonly used are linearity L_λ, planarity P_λ , sphericity S_λ, omnivariance O_λ, anisotropy A_λ, eigenentropy E_λ, sum Σ_λ of eigenvalues, and change of curvature C_λ. They are derived from normalized eigenvalues of covariance matrix, where $\lambda \geq \lambda_2 \geq \lambda_3 > 0$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Each of features is decribed with following formulas [8]:

$$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \text{ (1)}$$

$$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \text{ (2)}$$

$$S_\lambda = \frac{\lambda_3}{\lambda_1} \text{ (3)}$$

$$O_\lambda = \sqrt[3]{\prod_{j=3}^{3} \lambda_j} \text{ (4)}$$

$$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \text{ (5)}$$

$$E_\lambda = \sum_{j=3}^{3} \lambda_j \ln \lambda_j \text{ (6)}$$

$$\Sigma\lambda = \sum_{j=3}^{3} \lambda_j \text{ (7)}$$

$$C\lambda = \frac{\lambda}{\Sigma} \text{ (8)}$$

2.5. Classifiers

There are plenty types of classifiers available in machine learning area, though the most commonly used is Random Forest. Point clouds are quite big datasets what eliminates some solutions and also needs additional edition of training data[3]. In this paper random tree will be trained on 2 complete point clouds and tested on the other one to see how it works in case of unsupervised learning. Second chosen algorithm is Naïve Bayes for the comparison. Both were trained on the same dataset and tested on the same point cloud.

3. Experiment

Experiment was introduced to see the influence of two different neighbor acquisition methods. Two types of classifiers were used to check if the difference in results is only connected with certain type of classifier or has rather overall character. Procedure of the test was as follows:

1. Calculate Geometrical features per each point from Training clouds and testing cloud.
2. Train classifiers on training clouds
3. Classification with trained classifiers based on two classes: chair or not chair.
4. Visualize results and calculate correctness of predictions

All the steps were repeated for two types of neighborhood acquisition.

3.1. Experiment description

Testing was done with use of S3DIS database. Three different point clouds were used: office2 and office3 as learning database and office1 as the testing environment. Used annotations focused on chairs, therefore only two classes were considered. For the needs of experiment distinction was made only for chairs. IT means that recognition was based on following classes: chair, not chair. As it was already mentioned, for the classification Random Forest algorithm was used and for control purposes Naïve Bayes. Comparison of neighborhood acquisition methods focused on KNN and KNNR approaches, as those shown quite good performance. Finally, for the comparison in visual way, CloudCompare was used. Assumed K was 60 points and assumed range 6 cm.

3.2. Results

As the results, presented is visual comparison and percentage of correct guesses compared with labeled point cloud (red points represent class chair and blue not chair):

Figure 2: Naive Bayes classification with KNN

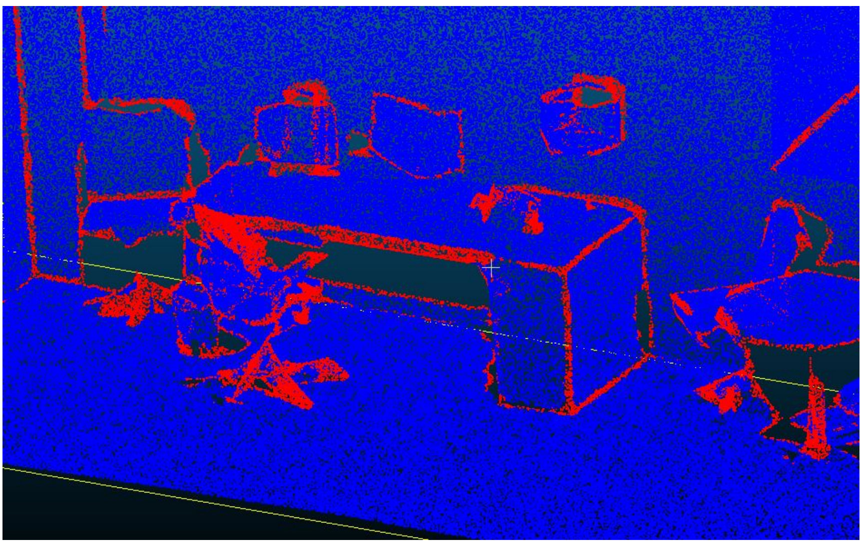

Figure 3: Naïve Bayes classification with KNNR

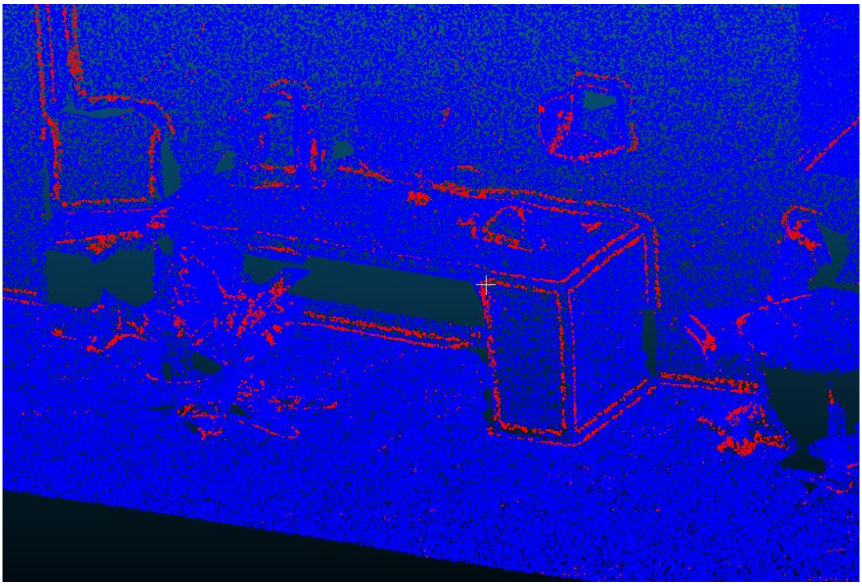

Figure 4: Random Forest classification with KNN

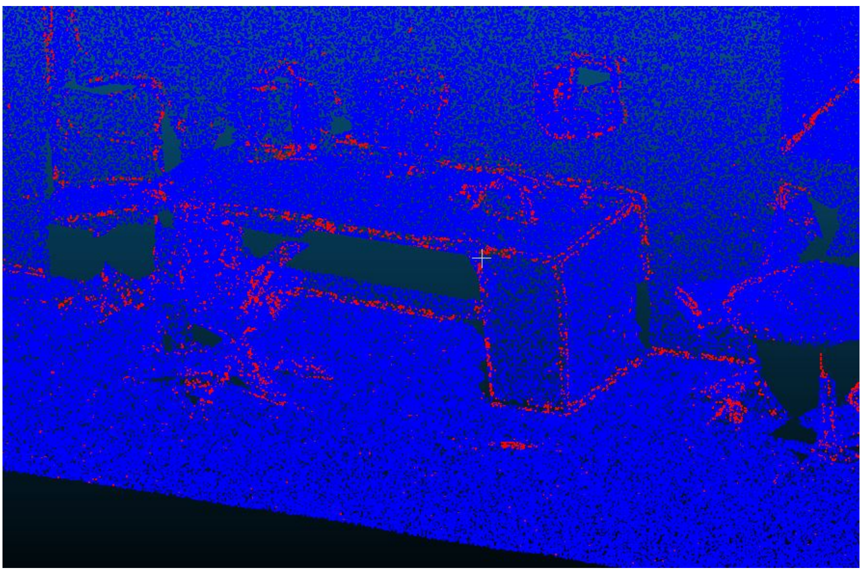

Figure 5: Random Forest classification with KNNR

Comparison of precision is shown below:

Method of classification	**Precision as percentage of correctly classifiedpoints in cloud**
Random Forest with KNNR	92.6816%
Random Forest with KNN	91.0828%
Naïve Bayes with KNNR	90.5434%
Naïve Bayes with KNN	89.5699%

3.3. Results discussion

From the results of the test, certain regularity can be noticed. For both classifiers precision increased for 1% in case of KNNR method. What was unexpected Naïve Bayes with lower precision allowed to outline in clearer way chairs, even though many other parts of the office were also classified as chair. Random Forest with better precision much less points classified as chair, therefore outline of chairs are not so easy to notice. For both classifiers, even though precisions seem to be quite high, based on visualization, there is quite big number of points that should be classified differently.

4. Conclusions

In this paper, some basic methods used for semantic segmentation of point cloud were overly discussed. Experiment was done to see if method of neighborhood acquisition has significant influence on classification precision. It is visible that for both classifiers KNNR methods gives better results. Even though, difference is only in 1%, for such big dataset it gives even 40 000 points classified correctly. For scale in 'office1' point cloud, all chairs consist of only 48 000 points. Therefore, 1% is a lot, based on data size. Additionally, Naïve Bayes with lower precision gave better outline of searched chairs. None of those methods allowed to label correctly all points assigned to chairs label, what in fact, gives some space for further tests and experiments. Perhaps, multi-class classification could give better results. Nevertheless, in comparison to different works in this area there was a try to classify types of objects based only on geometrical features, without usage of colors data. Keeping in mind this difference results seem to be considerably promising.

References

[1] Weinmann, M., *Reconstruction and analysis of 3D scenes*, Springer, 2016.

[2] Demantké, J., Mallet, C., David, N., and Vallet, B., *Dimensionality based scale selection in 3D lidar point clouds*, 2011.

[3] Thomas, H., Goulette, F., Deschaud, J.-E., and Marcotegui, B., *Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods*, 09 2018, pp. 390–398.

[4] Weinmann, M., Jutzi, B., and Mallet, C., *Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 2, No. 3, 2014, pp. 181.

[5] Landrieu, L. and Simonovsky, M., *Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs*, 06 2018.

[6] West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M., and Iverson, A. E., *Context-driven automated target detection in 3D data*, In: Automatic Target Recognition XIV, Vol. 5426, International Society for Optics and Photonics, 2004, pp. 133–143.

[7] Weinmann, M., Jutzi, B., Mallet, C., and Weinmann, M., *Geometric Features and Their Relevance for 3D Point Cloud Classification*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-1/W1, 06 2017.

[8] Weinmann, M., Weinmann, M., Mallet, C., and Brédif, M., *A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas*, Remote Sensing, Vol. 9, 03 2017, pp. 277.

[9] Li, M. and Sun, C., *Refinement of LiDAR point clouds using a super voxel based approach*, ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 143, 2018, pp. 213–221.

[10] Weinmann, M., Jutzi, B., and Mallet, C., *Feature relevance assessment for the semantic interpretation of 3D point cloud data*, ISPRS Annals of the

Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 5, No. W2, 2013, pp. 1.