5. Say Hello to Jetpack Compose and Compare with XML. URL: https://blog.kotlin-academy.com/say-hello-to-jetpack-compose-and-compare-with-xml-6bc6053aec13 (дата звернення: 19.11.2022).
6. Comparing Jetpack Compose performance with XML. URL: https://medium.com/okcredit/comparing-jetpack-compose-performance-with-xml-9462a1282c6b (дата звернення: 19.11.2022).
7. Jetpack Compose Stability Explained. URL: https://medium.com/androiddevelopers/jetpack-compose-stability-explained-79c10db270c8 (дата звернення: 19.11.2022).

DOI https://doi.org/10.30525/978-9934-26-277-7-161

LOW-COMPLEXITY LIDAR POINT CLOUD FILTERING METHOD FOR SELF-DRIVING VEHICLES

Matvienko V. T.
Candidate of Physico-Mathematical Sciences,
Associate Professor at the Department of Complex Systems Modelling
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine

Mushta I. A.
Graduate student at the Department
of Electronic Computational Equipment Design
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
Kyiv, Ukraine

Measurements obtained from LiDAR sensor always contain noise detections. The origin of this noise is different. Most LiDAR systems suffer degradation from adverse environment conditions. The photodetector of the lidar system detects transient light from the sun and the surroundings, and this light produces noise that hinders the system's effectiveness. Also adverse weather conditions, such as snow, dust, heavy rain or fog, distort the point cloud image obtained by LiDAR sensors. So to obtain high quality LiDAR point cloud filtering methods are used. Traditional filtering algorithms are often limited to isolated outliers, cannot identify outlier groupings or, otherwise, remove a lot of useful environmental features. What`s more, some of them are too complex to have ideal real-time performance. To adress these problems, this paper proposes new low-complexity LiDAR point cloud filtering method that is based on principal component analysis (PCA) and OPTICS-OF algorithm. Using the projection of original point cloud into the plane spanned by first principal component vector and second principal component vector of the correspondent covariance matrix we reduce time complexity of the algorithm. While using OPTICS-OF algorithm for clustering structure analysis and noise detection we assume spatial concentration of noise points.

Formulation of the proposed method

The key idea of the proposed method is similar to PCA-based dimension reduction method [1]. Input data dimension reduction is also used here to reduce overall time complexity. This approach is applicable here because of the input data pecularities. So noise points in most cases will not be shadowed by points that belong to useful data while projecting them on the plane.

Let`s assume that the number of 3D points in original point cloud is n. Then input points are firstly arranged into 3xn dimension matrix P_{3xn}. Each row of the matrix is zero-centered by substracting the means. Then the covariance matrix (1) is computed:

$$C = \frac{1}{n} P_{3\times n} P_{3\times n}^{T} \tag{1}$$

In the next step we find the eigenvalues and eigenvectors of the covariance matrix. Eigenvalues with corresponding eigenvectors are sorted in decreasing order. The first two eigenvectors that correspond to the first two eigenvalues are the vectors along which there is the greatest input data variance. So in the next step we project input point cloud into the plane spanned on the first two eigenvectors. Mathematically, the first two eigenvectors generate matrix T_{2x3}. Then the projection (2) is performed using the next formula:

$$Y_{2\times n} = T_{2\times 3} \times P_{3\times n} \tag{2}$$

The mechanism of this operation is shown on the figure 1.

For simplicity, we store core distance (4), local reachability distance (6) and outlier factor (7) for each point along with it`s coordinates in 2D projection (2) and 3D space. This gives us ability not to perform points restoration into 3D space at the end of the algorithm.

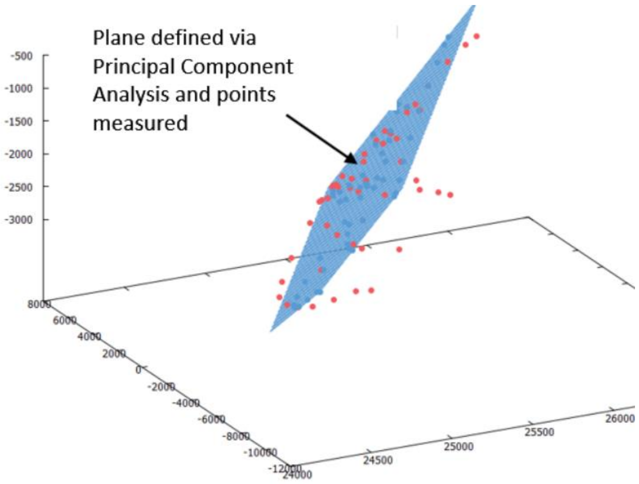

Figure 1. Projection of original point cloud into 2D plane

To filter obtained 2D point cloud it is useful to apply some kind of clustering, which is machine learning technique. Let`s consider the OPTICS algorithm for this purpose. OPTICS (ordering points to identify the clustering structure) is an algorithm for finding density-based clusters in spatial data [2]. OPTICS is a generalization of DBSCAN algorithm, used in [1], and overcomes it`s major weakness: the problem of detecting meaningful clusters in data of varying density. The key concept of this method is to order points in such a way that spatially closest points become neighbors. For each point, a special distance is stored that represents the density that must be accepted for a cluster so that both points belong to the same cluster. This is represented as a dendrogram, which is depicted in figure 2.

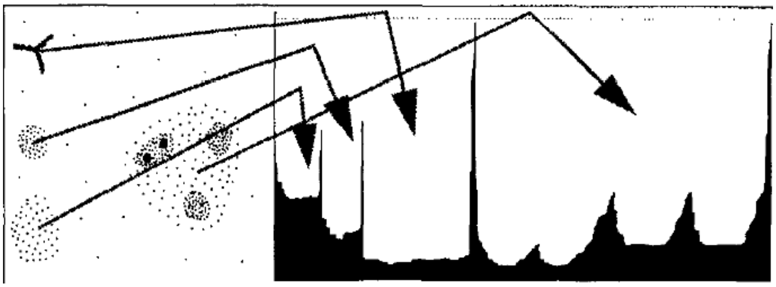

Figure 2. Dendrogram

Using this dendrogram not only traditional clustering information but also the intrinsic, hierarchical clustering structure can be extracted. But such extraction is not obvious and ambiguous. So let`s consider OPTICS-OF algorithm [3], which is a modification of OPTICS. The time complexity of this method in the simplest case is $O(n^2)$. To make it applicable to our task let`s use outlier detection method based on k-nearest neighbors-local outlier factor, which was proposed by He Xu, Lin Zhang, Peng Li and Feng Zhu for data filtration [4].

So, the next step of the proposed method is building KD-tree and computing *k*-nearest neighbors (3) for every point in 2D point cloud along with core-distance (4).

In mathematical notation:

For every point *y* from Y_{2xn} find:

$$N_{kdist(y)}(y) = \{y' \in Y_{2\times n} | d(y, y') \leq kdist(y)\} \quad (3)$$

Here $kdist(y)$ of y is the distance $d(y,o)$ between y and an object $o \in Y_{2xn}$ such that at least for k objects $o' \in Y_{2xn}$ it holds that $d(y,o') \leq d(y,o)$, and for at most k-1 objects $o' \in Y_{2xn}$ it holds that $d(y,o') < d(y,o)$ [14]. The set (3) is called *k*-nearest neighbors of *y*.

$$coredist_{\varepsilon,MinPts}(y) = \begin{cases} Undefined, if\ |N_{\varepsilon}(y)| < MinPts \\ MinPtsdist(y), otherwise \end{cases} \quad (4)$$

An object *y* whose ε neighborhood contains at least *MinPts* objects is said to be a core object. The core-distance (4) of object *y* is the smallest distance $\varepsilon' \leq \varepsilon$ such that *y* is a core object with respect to ε' and *MinPts* if such an ε' exists, i.e. if there are at least *MinPts* objects within the ε neighborhood of *y*. Otherwise, the core-distance is UNDEFINED [2].

In the second pass along the data for every *y* from Y_{2xn} we calculate reachability distance (5) $reachdist_{\infty,MinPts}(y,o)$ with respect to its neighboring objects $o \in N_{MinPts}(y)$ and local reachability distance (6) $lrd_{MinPts}(y)$ of y [2].

$$reachdist_{\varepsilon,MinPts}(y,o) = \begin{cases} Undefined, if\ |N_{\varepsilon}(o)| < MinPts \\ max\left(coredist_{\varepsilon,MinPts}(o), d(o,y)\right), otherwise \end{cases} \qquad (5)$$

$$lrd_{MinPts}(y) = \frac{1}{\frac{\sum_{o\in N_{MinPts}(y)} reachdist_{\infty,MinPts}(y,o)}{|N_{MinPts}(y)|}} \qquad (6)$$

In the last pass along the data outlier factor for every point *y* from Y_{2xn} is calculated (7) [2].

$$OF_{MinPts}(y) = \frac{\sum_{o\in N_{MinPts}(y)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(y)}}{|N_{MinPts}(y)|} \qquad (7)$$

To make noise filtration in 2D space, a threshold OF value should be chosen. There is an illustration of dependency between the density of the neighbors for each point and its OF value in the article [2]. This is shown in figure 3.

In figure 3 large OF values correspond to the points with small neighbors density and vice-versa. In the article [1] it was shown that in areas that are close to the sensor, points have high density and vice-versa. So the threshold OF value should be adjusted to the distance from the sensor to each point. This adjustment is done experimentally.

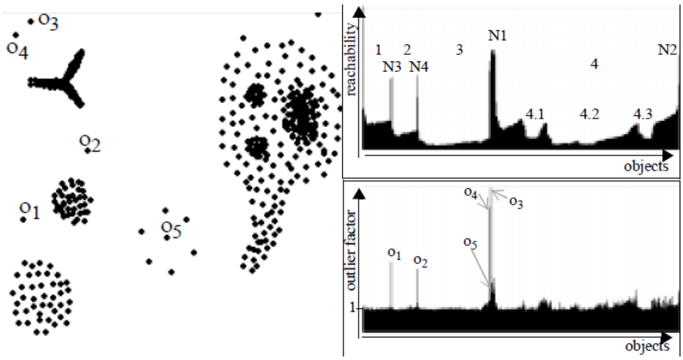

Figure 3. Depenency between the density of the neighbours for each point and it`s OF value

Conclusions

Point cloud filtering is a key step in building Advanced Driving Systems (ADS) and Advanced Driver Assistance Systems (ADAS) using LiDAR technologies. Low-complexity LiDAR Point Cloud Filtering Method for self-driving vehicles is proposed in this article. Low-complexity of this method is achieved by using data dimension reduction based on PCA, KD-tree usage and special structure for data storage. Compared with the traditional filtering algorithms, the proposed method reduce the overall time complexity and preserve more environmental features while filtering noise points.

References:

1. Y. Duan, C. Yang, H. Chen, W. Yan, H. Li Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction Opt. Cmmun., 482 (2021), Article 126567.
2. Kriegel, Hans-Peter; Kröger, Peer; Sander, Jörg; Zimek, Arthur (May 2011). "Density-based clustering". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 1 (3): 231–240.
3. Breunig MM, Kriegel H-P, Ng RT, Sander J (1999) Optics-of:Identifying local outliers. In: *European conference on principlesof data mining and knowledge discovery*, pp. 262–270. Springer, Berlin.
4. He Xu, Lin Zhang, Peng Li, and Feng Zhu. 2022. Outlier detection algorithm based on k-nearest neighbors-local outlier factor. *Journal of Algorithms & Computational Technology* 16(2022), 17483026221078111