Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/19/20231018

DBSCAN-based line density clustering algorithm for CAD architectural drawings

Rui Li

School of Construction Engineering, Jilin University, Changchun, Jilin Province, China, 130000

lruishaoxia@gmail.com

Abstract. In the traditional architecture industry, architects often need to convert CAD drawings into BIM in order to express the final effect of the building more concretely and to understand whether the design of each profession is reasonable after the drawings are completed. The whole modeling process is boring and tedious, and due to human fatigue, the final result is prone to problems, which eventually leads to failure to meet expectations. In order to free architects from the tedious task of model transformation, companies have designed software for automatic model transformation using computers. The core of the software design is related to computational geometry. Line segment clustering is one of the elements of computational geometry and a frequent problem in programming. Appropriate clustering of line segments can often bring convenience to the problem. This paper combines the basic idea of the DBSCAN algorithm, improves the shortcomings of DBSCAN algorithm in dealing with line segment clustering, and proposes a density-based line clustering algorithm with shapely library as a tool. The algorithm is highly interpretable, and the method performs well and efficiently in the test of actual drawings, whether to group the line segments or to find the target line segments that meet the conditions, which provides convenience for the subsequent calculation.

Keywords: DBSCAN, linear density clustering, machine learning, BIM.

1. Introduction

Autodesk invented BIM (Building Information Modeling) technology in 2002, and as a result of its considerable technical benefits such high accuracy simulation, high resource coordination, and full life cycle management, it is now widely acknowledged in related sectors throughout the world. [1, 2]. BIM is three-dimensional data, which can better visualize the data value. CAD architectural drawings are two-dimensional data. Manual drawing is more convenient and the number of drawings in the industry is greater than that of BIM. If the use of computer automation to convert two-dimensional CAD drawings to BIM models, it will bring great convenience to the construction industry [3]. This 2-D to 3-D software design often involves the basic geometry of similar points and lines, and clustering of lines is a problem that cannot be avoided when dealing with 2-D drawings.

The clustering algorithm is an important part of machine learning, and researchers have proposed many clustering algorithms, such as K-means, DBSCAN, OPTICS [4, 5]. and when clustering lines, most of the time it is the lines that need to be densely clustered. The DBSCAN algorithm is to use the density connectivity of clusters to discover arbitrarily shaped clusters. The algorithm has a natural

DOI: 10.54254/2755-2721/19/20231018

resistance to noise, but it is only applicable to spatial data with point-like samples, and there is no way to apply it to line segments. Therefore, this paper proposes a density clustering algorithm for lines based on the density-based idea of DBSCAN, which has good results after experiments.

This paper proposes a line density clustering algorithm based on the density-based idea of DBSCAN, which has good results and facilitates the geometric calculation in the process of converting CAD data to BIM data.

2. DBSCAN

2.1. Basic idea

The DBSCAN algorithm is a density-based clustering algorithm [6]. The basic step is to iterate through each point and determine the number of other points in a specific circular region around it (influenced by the domain radius Eps). If the number of other points in the region is greater than a set value (MinPts), then the point is labeled as a core point and given a cluster number. The above operation is repeated for the other points in the area. The region will slowly expand as the steps proceed, and the core points will be more and more numerous, and these core points are marked with the same cluster number for the same cluster. When the points in a region are traversed, the above operation is repeated for any of the remaining points that are not traversed, until all the points are traversed. At this point, the marked points will be divided into different clusters, and the points that are not marked into arbitrary clusters are noise.

2.2. Algorithm deficiencies

From the operation of the DBSCAN algorithm and its parameters (radius of the circular region: Eps, a minimum number of points in the region: MinPts), it can be concluded that the data samples of the DBSCAN algorithm are limited to point samples only, because in a sense a circle is a point. This definition of the perimeter of a point can be described by a circle, but the definition of the perimeter of other geometries (e.g., lines, polygons) is clearly inappropriate to be described by a circle.

Then, from another parameter, the minimum number of points in the region (MinPts), for the point set, the points in the region of the point must be an integer number of points. There is no case of a half-point, so this parameter is reasonable for the point set. However, for line set and polygon set, there are half lines or half polygons in a region, so it is not reasonable to describe by number alone.

In summary, the DBSCAN algorithm is not suitable for line clustering, but if the above two points are optimized according to the characteristics of lines, then a density-based line clustering algorithm can be obtained.

3. Density-based line clustering algorithm

3.1. Basic idea of migrating DBSCAN

The key to the density clustering algorithm is understanding density. Density is defined in physics as the number of objects per unit area. For points, the more points per unit surface set means the denser the points at the position, the more they can be gathered together. Referring to Figure 1, from the human point of view, the intuition is to divide the many lines in the diagram into six regions. Then, from the perspective of a single line, describing whether the area around a line is dense or not can be described by the total length of the lines in the area around that line (excluding itself). This brings us to the two basic concepts of DBSCAN-area and number.

DOI: 10.54254/2755-2721/19/20231018

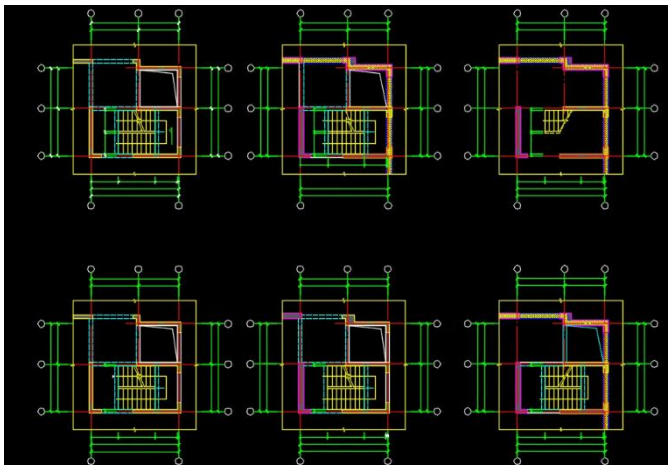

Figure 1. CAD drawing of a building stairway.

First, there is the issue of regions. In analogy to the point region being a circle, the line region can be thought of as the region formed by the line's buffer operation in Python's shapely library. In contrast to the radius of the point region, the "radius" of the line region can be thought of as the distance parameter of the buffer function, as shown in Figure 2, where both the green and yellow regions are the product of buffering a certain value for the corresponding line. It can be considered that the region is some kind of enlarged form of the object. The next issue is the number of regions. The number of points in the point region is the number of points, analogous to the length of other lines in the line region.

Figure 2. Line set.

3.2. Algorithm definition

Definition 1: Region of a line segment--line_area.

In this paper, the surrounding rectangular area is obtained after performing geometric operations on the line segment, as shown in the green and yellow boxes in Figure 2.

Definition 2: Core Line Segment.

If the ratio of the sum of the lengths of the other lines in the line segment area to the length of the line segment exceeds a certain set value, then it is a core line segment, and the line segment is divided into specific clusters.

3.3. Algorithm parameters

Parameter 1: Proportion of line extension--length_ratio.

The line is initially expanded by a certain amount because the area surrounding the line's terminal is also taken into account. Since it is obviously not reasonable to extend the line by a fixed distance for all lengths, the extension distance is a certain ratio of the line length. The ratio is length_ratio, and the extension of line l1 in Figure 3 is l.length*length_ratio.

DOI: 10.54254/2755-2721/19/20231018

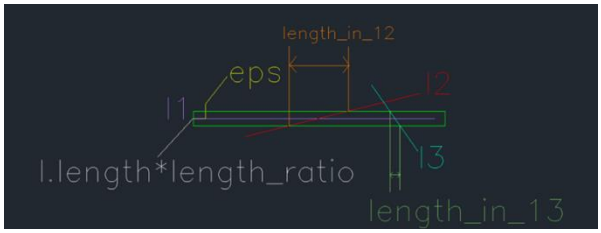

Figure 3. Parameter description.

Parameters 2: Parameter of the line buffer—eps.

Any line segment is extended and buffered with eps as a parameter (l.buffer(eps, cap_style=2)) to form a region.

Parameter 3: Ratio of the sum of the minimum line lengths in the region--min_length_ratio.

This parameter is analogous to the DBSCAN parameter MinPts, but it cannot be a fixed value for lines. For example, in Figure 4, if the combined length of the other lines in the line area is considered to be a fixed value such as 100, then this may result in l1 also being a core line segment and l5 also being a core line segment. However, l1 is not supposed to be combined with the others, so this parameter should be a dynamic one, related to the original lines of the region.

Therefore, if you want l1 to be a core line segment and to be grouped with other lines, the minimum value of the ratio of the sum of the lengths of the other lines in the region of l1 to the length of l1 itself is required. The condition for the above to hold is (length_in_12+length_in_13)/l1.length>= min_length_ratio.

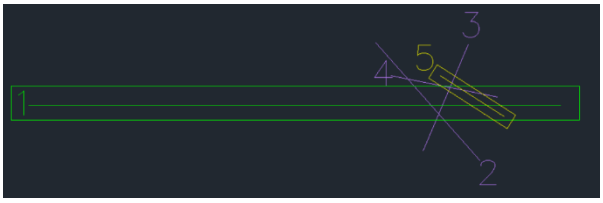

Figure 4. Line set.

3.4. Algorithm description

3.4.1. Flowchart.

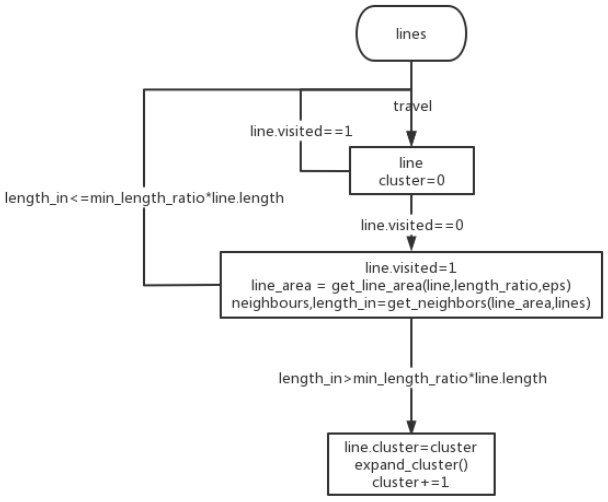

Figure 5. Flow chart of the algorithm.

DOI: 10.54254/2755-2721/19/20231018

3.4.2. Description. **Preceding attributes:**

cluster=0, 0 means noise, positive number means cluster

line.visited=0, 0 means not visited, 1 means already visited

line.cluster=0

The algorithm scans every line segment object in the line set in the plane. If the line_visited==1 of the line segment, it means that the line segment has been classified, then skip the line segment. If the line_visited==0 of the line segment, it means that the line segment has not been classified. Set the line segment according to the parameters, extend the length_ratio*line.length at both ends, and then use the buffer function of the shapely library to expand the extended line according to the set parameters , thus forming the line area.

The line segment is extended at both ends according to the parameter setting (length_ratio*line.length), and then the extended line is expanded using the buffer function of the shapely library according to the set parameters (eps) to form the line area (line_area).

Then use the get_neighbors function, which uses the intersection function in the shapely library, to find the line segments (other than itself) covered by line_area and calculate the sum of their lengths (length_in). If length_in>min_length_ratio*line.length, then the line segment is a core line segment, line.cluster=cluster.

Then this study carried out the operation of expanding the cluster (expand_cluster()). The steps are similar to the above-mentioned finding the core line segment: traverse the neighbors. If line.visited==1, then skip; otherwise continue to judge whether it is a core line segment, If yes then line.cluster = cluster, otherwise noise. Until line.visited==1 for all lines in neighbors, then cluster+=1.

Repeat all the above operations until all lines are visited.

3.5. Practical test results

Test 1. Figure 6 is a CAD drawing, and Figure 7 is obtained through data analysis. In this study, the parameter length_ratio is set to 1/80, eps is set to 500, min_length_ratio is set to 0.9, and the linear density clustering algorithm is used to obtain the results in Figure 8. Finally, for each cluster, limit the number of lines in the cluster, which can simply divide the original plane of the lines to be processed into six categories.

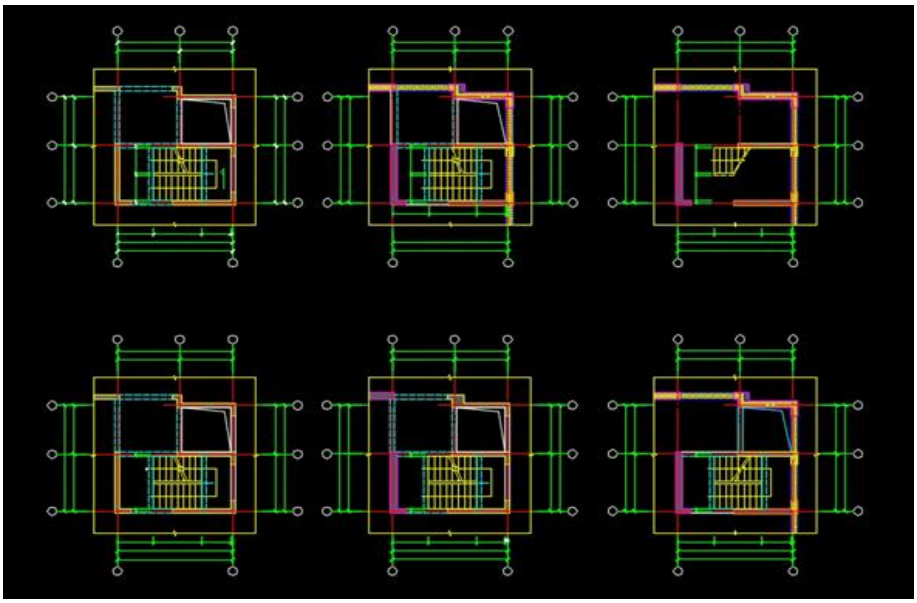

Figure 6. CAD drawing of a building stairway.

DOI: 10.54254/2755-2721/19/20231018

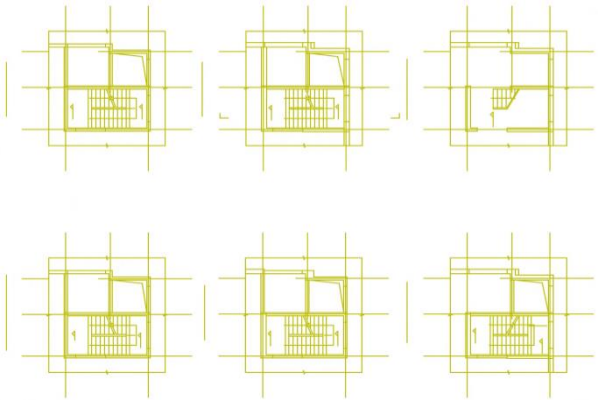

Figure 7. Parsed line data.

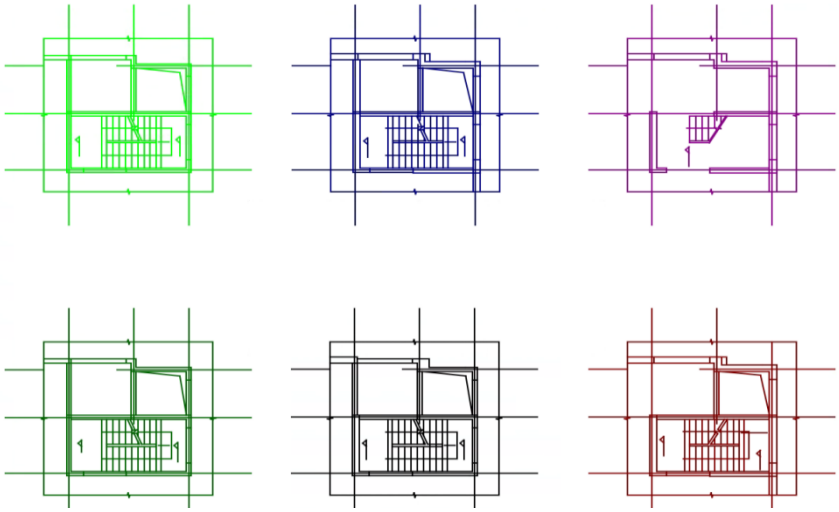

Figure 8. Line_clusters.

4. Conclusion

In this paper, based on DBSCAN, a density-based line clustering algorithm is proposed for the characteristics of lines. The algorithm can cluster and group lines according to different needs in the face of different data sets, and the obtained results can make the subsequent calculation more convenient. It is also possible to select the lines that have been filtered to get the target lines. Two practical use cases have shown that the algorithm can bring great convenience to the geometry part of the programming process and thus to the development of automatic transformation model software. Since this algorithm is a DBSCAN-based machine learning algorithm, it is highly interpretable. In addition, due to the setting of the relevant parameters, the algorithm is theoretically applicable to the clustering of curves as well and has good generalizability.

References

[1] Martins J P,Abrantes V.Automated code-checking as a driver of BIM adoption[J].Journal for Housing Science.2010,34(4):286-294.

[2] LI M,LIU N.Practice of BIM Teaching Scheme for Application Rriented Talents Training[J]. The International Journal of Electrical Engineering&Education,2021:0020720921996607.

[3] Lirong Guo, Liang Wang. Research of Mechanical and Electrical Engineering Quantity Calculation Based on Revit Secondary Development[J]. Journal of Information Technologyin Civil Engineering and Architecture, 2023, 15(1): 30-36. doi: 10.16670/j.cnki.cn11-5823/tu.2023.01.06

DOI: 10.54254/2755-2721/19/20231018

[4] XU D K, T Y J. A comprehensive survey of clustering algorithms[J]. Annals of Data Science, 2015, 2(2): 165-193.
[5] Ankerst M, Breunig M M, Kriegel H P, et al. OPTICS: ordering points to identify the clustering structure [C] // SIGM0D 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, Usa. 1999:49-60.
[6] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]. AAAI Press, 1996.