A Scalable Open-Source System for Segmenting Urban Areas with Road Networks

Ming Zhang
Business Intelligence Lab, Baidu Research, China
zhangming20@baidu.com

Yanyan Li
Business Intelligence Lab, Baidu Research, China
liyanyanliyanyan@baidu.com

Jianguo Duan
Business Intelligence Lab, Baidu Research, China
duanjianguo@baidu.com

Jizhou Huang
Baidu Inc., China
huangjizhou01@baidu.com

Jingbo Zhou*
Business Intelligence Lab, Baidu Research, China
zhoujingbo@baidu.com

ABSTRACT

Segmenting an urban area into regions is fundamentally important for many spatio-temporal applications. The traditional grid-based method offers a simple solution as it divides the city map into equal-sized grids, but it fails to preserve semantic information about the original urban structure. Several studies apply the road network to cut the metropolitan area into meaningful blocks. However, existing works do not achieve good scalability, and there is no public system provided so far. To address those problems, we build *GENREGION*, the first scalable vector-based system which generates reasonable regions using all levels of the road network. We conduct an evaluation to prove the efficiency and effectiveness of our system. We also publish our system as a Python library through Python Package Index (PyPI), and demonstrate its utility using real public datasets in this paper. The source code and useful instructions can be found on https://github.com/PaddlePaddle/PaddleSpatial/tree/main/paddlespatial/tools/genregion.

CCS CONCEPTS

• **Software and its engineering → Software libraries and repositories**; • **Information systems → Clustering**.

KEYWORDS

Urban Area Segmentation, Road Network, Vector-based Model

ACM Reference Format:
Ming Zhang, Yanyan Li, Jianguo Duan, Jizhou Huang, and Jingbo Zhou*. 2024. A Scalable Open-Source System for Segmenting Urban Areas with Road Networks. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3616855.3635703

*Jingbo Zhou is the corresponding author.

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635703

1 INTRODUCTION

Segmentation of urban area has been widely demanded as the prerequisite for many spatio-temporal applications such as urban region analysis [7], trajectory data mining [5] and traffic prediction [3]. Many researchers instinctively choose the grid-based method in their studies to segment a map into equal-sized rectangle grids [2], which have little semantic meaning to the urban structure.

To generate useful regions while retaining city structure, urban road networks offer a promising solution. These networks inherently convey public transportation, human mobility, and urban economic activity, delineating work and residential areas. Thus, interconnecting road segments can divide the map into semantic regions. Also, city road network data is readily available on platforms like OpenStreetMap [1] and public repositories such as Figshare.com [4]. Consequently, road network-based segmentation provides a feasible solution with two primary models: raster-based and vector-based, for geospatial analysis and map segmentation.

The raster-based model divides an area into discrete grid cells, representing road segments as sequences of these units. Yuan et al. [6] introduced a raster-based approach for urban area segmentation using the road network. They employ morphological techniques like dilation and thinning, to simplify the road map. Subsequently, connected component labeling identifies connected pixels, forming regions. Nevertheless, the model's accuracy hinges on image quality when discretizing the road network. Additionally, since both the input and output of this model are images, it poses challenges for researchers aiming to conduct geospatial analysis requiring topological and mathematically-friendly region objects.

The vector-based model employs geometric primitives (points, lines, polygons) for spatial objects, offering accurate and mathematically intuitive representations. These representations facilitate tasks like finding shortest paths and conducting topological analyses. Zhao et al. [8] use a vector-based approach to segment urban areas via the road network. They simplify the graph by merging points within a threshold and recursively apply the Dijkstra algorithm until the region becomes inseparable. Although this algorithm addresses raster-based model issues, it introduces challenges in boundary identification and scalability. Efficient border identification remains undisclosed, and even with pre-prepared boundary information, segmenting an entire urban area is unfeasible due to the algorithm's super-linear complexity concerning the number of nodes in a complex road network.

In this paper, we release *GENREGION* - a practical system with releasing an open-sourced Python Library for segmenting urban areas using road networks through a vector-based approach. We also demonstrate its utility using a public real-life dataset. To the best of our knowledge, the *GENREGION* system is the first scalable vector-based approach to generate regions based on all levels of the road network.

The proposed approach views the urban road network as a graph, where each road segment becomes an edge with its two endpoints as vertices. We streamline the graph by grouping vertices based on a novel distance metric and break each edge at intersection points to maintain graph connectivity. Then, we introduce a novel "leftness" indicator for each road segment and create regions by iteratively identifying the leftmost vector of an edge until it forms a closed loop. Finally, we amalgamate small blocks and eliminate sub-regions to produce a concise list of meaningful polygons.

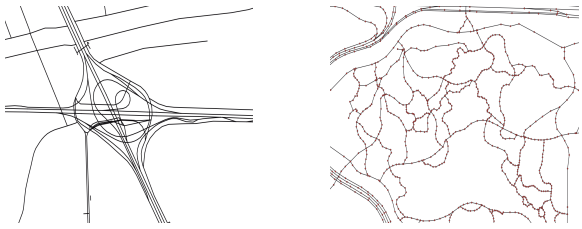
To summarize, we make these contributions:

- We significantly improve the scalability of the vector-based map segmentation algorithm using the road network by finding the leftmost vector recursively.
- We enhance the semantic level of the generated regions through a new distance metric while clustering during graph simplification.
- We release the first open-sourced system for region generation using the road network, and demonstrate its utility using a real dataset.

2 SYSTEM OVERVIEW

This section clarifies our *GENREGION* system after we preprocess the data into a list of road segments. It consists of three major processing components to generate desired polygons: I) Simplifying segments based on clustering; II) Generating closed polygons and III) Polygons refinement.

2.1 Simplifying Segments Based on Clustering



(a) Interchangeable Roads **(b) Curvy Roads**

Figure 1: Intricate Road Network

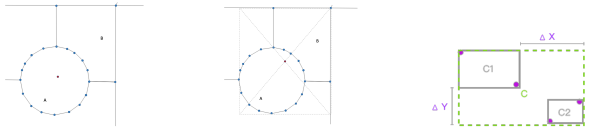
The urban road network often has extraneous details that hinder region generation. Interchangeable roads (Fig. 1a) can produce redundant blocks, while curvy roads (Fig. 1b) or rings with many sub-segments strain computations. To simplify the map, we use a hierarchical clustering algorithm to decrease segment and node counts.

The clustering algorithm groups nodes by location similarity. Essentially, we simplify the graph by choosing each cluster's center as its representative point and then reconnecting these nodes to restore segments. Zhao et al. [8] used this method, calculating Euclidean distances between node pairs for segmentation. But, due to vector-based geospatial primitives' nature, this introduces bias. The vector model represents curves with multiple straight lines, meaning a zigzag road needs more nodes than a straight one of similar length. If we cluster curved and straight roads based on Euclidean distance, the weighted mean of their nodes will lean more towards the curves than the straight paths (Fig. 2a).

To eliminate this bias, we adopt a new means to find representative points in a cluster and a novel distance metric to measure spatial proximity. For each cluster, we record maximum and minimum x, y-coordinates of its members (we will call them MBRs later) and generate a representative rectangle based on these values. The new representative point of a cluster will be the center of its representative rectangle, and it will no longer be influenced by the number of nodes of a particularly curvy road type (Fig. 2b).

To directly measure the difference between MBRs of two clusters, we redefine the distance metric for clustering nodes in the same way as finding the representative node. Eq. 1 and Fig. 2c demonstrate our elaboration thoroughly. To obtain a simplified graph, we assign each node as an initial cluster and hierarchically cluster nodes with the well-defined distance metric. If the distance between two targets is less than a given threshold, we combine sets of nodes in both clusters and calculate the new MBR information.

By dividing the map into equal-sized grids and noting each node's location, we enhance algorithm efficiency. To identify mergeable cluster candidates, there's no need to sift through all clusters. We only inspect clusters in the same or neighboring grids as the target cluster, as they're the closest geographically. After merging two clusters, we gather their connectivity data, assign grid info, and remove them from consideration to avoid redundancy. The hierarchical clustering ends when all cluster pairs exceed a distance threshold. Then, we simplify the road network by connecting each group's representative points using the retained connectivity information.



(a) Original Center **(b) MBR Center** **(c) Cluster Distance**

Figure 2: Distance Metric

$$d(c_1, c_2) = \begin{cases} \infty & \text{if } c.x > \mathrm{T} \text{ or } c.y > \mathrm{T} \\ (c.x - c1.x) + (c.y - c1.y) & \text{if } c1.x + c1.y \geq c2.x + c2.y \\ (c.x - c2.x) + (c.y - c2.y) & \text{otherwise} \end{cases} \tag{1}$$

2.2 Generating Closed Polygons

We adopt the same grid-based indexing technique to break segments into sub-segments by their intersections. This guarantees that each road link only connects to others at its two end nodes. After breaking those segments, we make each segment into two

vectors which point in opposite directions, as a side can be used twice at most by two adjacent regions. The *invec* (*outvec*) of a node represents the vector that ends (starts) at it, and the *invec* (*outvec*) of a vector stands for the vector that points to its start node (begins at its end node). The leftness of a vector's *outvec* is another new concept. Ostensibly, we clarify that a left *outvec* of a vector sits at its left side, and we also formally present a precise mathematical definition for it (Eq. 2). Besides, we mark that an *outvec* is on the left if it has the same direction as the vector. We also design a way to calculate the value of the leftness with the help of the trigonometric properties (Eq. 3), and the *outvec* with the smallest leftness value is the leftmost *outvec*.

$$\text{side(vec, } outvec) = \begin{cases} \text{left} & \text{if } sin(vec, outvec) > 0 \text{ or} \\ & cos(vec, outvec) = 1 \\ \text{right} & \text{otherwise} \end{cases} \tag{2}$$

$$\begin{aligned} \text{side} &= \text{side(vec, outvec)} \\ \text{cos} &= \text{cos(vec, outvec)} \\ \text{leftness} &= \text{side} * \text{cos} + (1 - \text{side}) \end{aligned} \tag{3}$$

Our *GENREGION* system creates regions by continually identifying the leftmost outvec for each vector. Every vector is used once, and marked as "used" after forming part of a polygon. For every untouched vector, we identify its leftmost outvec and continue this for subsequent *outvec* until the end node of one aligns with the original vector's start node. This process defines a polygon's boundary, forming a region. This approach efficiently creates polygons, ensuring the resulting polygon is the smallest to a vector's left side. By pinpointing every valid vector's leftmost outvec, we swiftly segment the map into a list of polygons.

2.3 Refining Polygons

Though usable polygons are generated in the previous step, there are still two deficiencies needed to be improved, sub-polygons and tiny/slender polygons. Sub-polygons represent areas already contained in larger polygons, and a tiny polygon cannot characterize a meaningful region solely. A slender polygon refers to a long and narrow area which typically stands for a highway road.

The grid-based indexing technique catalyzes the computation once more while searching for those unwanted polygons. Sub-polygons can be detected by comparing their MBR information to polygons in the same or nearby grids, and we will directly remove them. Tiny and slender polygons are identified by thresholds of the area and the width separately. We define the width of the polygon as the ratio of its area to the perimeter, for the shape of a polygon can be versatile. A tiny or slender polygon will generally melt with the abutting region that shares the longest common segment. With those undesired polygons handled, the ultimate output is delivered to users as a list of well-defined Python Shapely polygons.

3 EVALUATION

A competent urban map segmentation algorithm should generate regions with enough semantic information within a fair amount of time. Therefore, we conduct experiments concerning these two parts in the following paragraphs.

To measure the semantic level of the result, we use regions of interest (ROI) for cities in China on Baidu Maps. These ROIs contain boundary coordinates of important regions in a city, like hospitals, shopping malls and schools. Thus, a high Jaccard similarity (ratio of the intersection to the union) between a generated region to an ROI indicates a meaningful segmentation. We compare the result of our model to others using the road network of Beijing, Shanghai, Shenzhen, Chengdu and Wuhan, located in the northeast, east, southeast, southwest and middle of China, respectively. We replicate the methods of Yuan et al. (*mapseg*) [6], Zhao et al. (*shortpath*) [8] and grid-based segmentation in 100, 500 and 1000 meters separately. Table 1 presents the evaluation result, and our system achieves the best performance. Note that an average of 40% Jaccard Similarity is comparatively high considering that ROIs does not contain the portion of road. Compared to traditional grid-based methods, urban map segmentation using the road network significantly improves the semantic level of segmented regions.

Average Percent of Jaccard Similarity					
Model Name	Beijing	Shanghai	Shenzhen	Wuhan	Chengdu
GENREGION	41.0%	42.7%	41.1%	32.8%	38.4%
shortpath	39.3%	40.1%	38.8%	31.8%	37.0%
mapseg	33.4%	39.1%	30.9%	31.7%	37.5%
Grid-100	0.6%	0.3%	0.8%	0.5%	0.7%
Grid-500	11.8%	10.7%	10.9%	9.4%	10.4%
Grid-1000	8.2%	6.6%	6.3%	6.7%	6.3%

Table 1: Algorithms performances on different cities

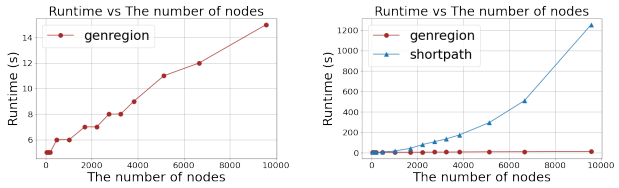


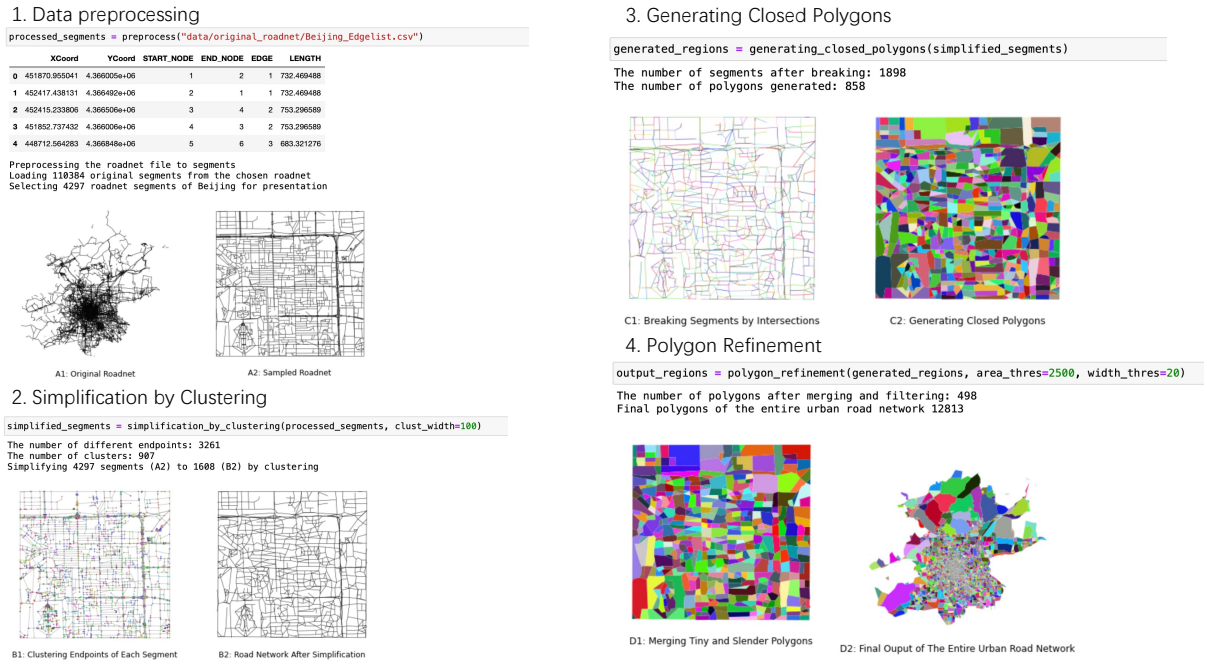
Figure 3: Scalability

We also conduct experiments on the scalability of two vector-based systems in road network with different levels of complexity. The complexity of an urban road network is typically characterized by the number of nodes or the number of segments. In Fig. 3, the left graph shows the run time of GENREGION as the number of nodes increases (same trend when using segments), and the right graph compares the scalability of GENREGION and *shortpath*. We can see that the extraordinary scalability of GENREGION outperforms the scalability of *shortpath*.

4 DEMONSTRATION SCENARIO

Here we walk through an elaborate demonstration (Fig. 4) about *GENREGION*. We show potential users the entire procedure of generating desired regions interactively.

Step 1. The data preprocessing step converts the raw road network dataset to a list of segments. The road network dataset selected in the demo is a public dataset of Beijing from Figshare.com [4]. The "preprocess" function takes the dataset path as the input. It aggregates points that belong to the same road and generates 110,384 original road segments. In this demo, we also select a subset set of



	XCoord	YCoord	START_NODE	END_NODE	EDGE	LENGTH
0	451870.955041	4.366005e+06	1	2	1	732.469488
1	452417.438131	4.366492e+06	2	1	1	732.469488
2	452415.233806	4.366506e+06	3	4	2	753.296589
3	451852.737432	4.366006e+06	4	3	2	753.296589
4	448712.564283	4.366848e+06	5	6	3	683.321276

Figure 4: Demonstration of The Region Generation Process

4,297 road segments so that users can clearly understand what is happening.

Step 2. The "simplification_by_clustering" function performs the clustering process, taking original segments and cluster width as input. The sampled network has 3,261 nodes, and our clustering methods reduce the number of nodes to 907. In the graph, points with different colors belong to distinct clusters, and their MBR centers were used as the representation points. After reconnecting segments to those new nodes, we reduced the number of segments to 1,608.

Step 3. In this step, the "generating_closed_polygon" function first breaks road segments by their intersections. Then, our algorithm generates 858 polygons by recursively finding the leftmost *outvec* for each vector.

Step 4. The final step calls "polygon_refinement" function, which takes the input of the area threshold and the width threshold. Polygons with area or width (ratio of the area to the perimeter) less than the corresponding threshold will be merged into their neighborhood. The function also remove sub-polygons to get rid of the redundant information. Finally, we get 498 polygons from the chosen Beijing road network and 12,813 total polygons from the entire network.

Except for the step-by-step demonstration, we also prepare an end-to-end function called "generate_regions" which aggregates the last three steps and converts a processed road network to a list of polygons directly in our Python package.

5 CONCLUSION

In this paper, we present an open-sourced scalable system to segment urban areas using road networks. We also demonstrate the effectiveness and efficiency of our system.

ACKNOWLEDGMENTS

The work was supported in part by the National Natural Science Foundation of China under Grant No.92370204.

REFERENCES

[1] Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive computing* 7, 4 (2008), 12–18.
[2] Jason W Powell, Yan Huang, Favyen Bastani, and Minhe Ji. 2011. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *International Symposium on spatial and temporal Databases*. 242–260.
[3] Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. 2020. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).
[4] Mike Thelwall and Kayvan Kousha. 2016. Figshare: a universal repository for academic resource sharing? *Online Information Review* (2016).
[5] Ming Xu, Jianping Wu, Mengqi Liu, Yunpeng Xiao, Haohan Wang, and Dongmei Hu. 2018. Discovery of critical nodes in road networks through mining from vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems* 20, 2 (2018), 583–593.
[6] Nicholas Jing Yuan, Yu Zheng, and Xing Xie. 2012. Segmentation of urban areas using road networks. *Microsoft, Albuquerque, NM, USA, Tech. Rep. MSR-TR-2012-65* (2012).
[7] Jichang Zhao, Ruiwen Li, Xiao Liang, and Ke Xu. 2015. Segmentation and evolution of urban areas in Beijing: A view from mobility data of massive individuals. In *The 12th International Conference on Service Systems and Service Management*. 1–6.
[8] Si Zhao, Hongwei Wu, Lai Tu, and Benxiong Huang. 2014. Segmentation of Urban Areas Using Vector-Based Model. In *IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and IEEE 11th Intl Conf on Autonomic and Trusted Computing and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*. 412–416.