Sensors and Materials, Vol. 36, No. 10 (2024) 4441–4453
MYU Tokyo

S & M 3810

Method of Extracting Intersection Polygons Based on Centerline Junctions

Lei Cai,[1,2] Yong Yin,[3*] Qifei Mao,[3] and Jiannan Yang[3]

[1]Beijing Institute of Surveying and Mapping, Beijing 100038, China
[2]Beijing Key Laboratory of Urban Spatial Information Engineering, Beijing 100038, China
[3]Chinese Academy of Surveying & Mapping, Beijing 100036, China

(Received July 26, 2024; accepted October 11, 2024)

Keywords: intersection extraction, centerline, topology, nearest points, polygon split

Intersection polygons are an important component of road data. To address the challenge of extracting the microstructure of roads, we propose a method of extracting intersection polygons based on centerline junctions. The method involves the following steps: (1) topological preprocessing of the road centerline to extract dangling nodes and converting road polygon data into line rings, (2) matching the nearest line ring within a certain threshold range of dangling nodes to obtain splitting points, (3) calculating the set of splitting points associated with the line rings and dividing the line rings to obtain boundary arcs, (4) extracting the boundary arcs associated with the current intersection from the intersection nodes of the road centerline, and (5) calculating the primary nearest point set and the secondary nearest point set in turn, and using the secondary nearest point set to extract the resulting intersection polygons. Experimental results indicate that the method can effectively generate the required intersection polygons at road intersections.

1. Introduction

Road intersections are common and important hubs in traffic networks, as well as important components of urban road systems.(1) With the development of road construction, the crisscrossing network becomes increasingly complex. In the basic geographic information data, road intersection is one of the important road microstructures. Nowadays, studies on the automatic extraction of road intersections are numerous. In terms of big data, studies such as those described in Refs. 2–5 identify turning point clusters and their central points with vehicle track data, use different spatial clustering methods to extract intersections, and identify various intersection structures. In terms of image processing, Li *et al.*(6) proposed a method of extracting intersections from high-resolution images by using feature semantic rules. Chen *et. al.*(7) proposed a quantitative mapping relationship based on the pixel and intersection structures, providing feature semantic information to locate the intersection center and intersection structure. There are also similar research studies on extracting intersections based on image

*Corresponding author: e-mail: yinyong@casm.ac.cn
https://doi.org/10.18494/SAM5252

ISSN 0914-4935

information[8,9] and extracting road intersections from point cloud data.[10] In practical applications, intersections, which are the necessary places vehicles converge, turn, and exit, are the areas with high incidence of traffic accidents. Studies of, for example, vehicle control,[11] factors affecting the severity of single-vehicle accidents,[12] crowd evacuation under emergency,[13] and the diffusion law of particulate matter in air at urban road intersections[14] have been conducted. All of them demonstrate the importance of road intersections in road networks. There are other studies such as road planning outcome information management,[15] supervised learning with clustering to complete intersection identification,[16] and automatic intersection detection in 3D scenes.[17] However, there are few research studies on intersection extraction based on 2D planar vector road surface data. An intersection extraction method based on the road centerline and road surface is proposed in this study. There are two core steps: (1) splitting the road surface based on the feature relationship between topological nodes and the road surface and (2) locating the intersection nodes, returning to the intersecting part of the road under its buffer with the split road, and calculating the information of multiple nearest points from the intersection to the intersecting line segments, and then combining the eligible nearest points to enclose and synthesize the intersection surface.

2. Extraction of Intersection Polygons Based on Centerline Junctions

2.1 Basic principle

In this paper, we propose a method for extracting intersection polygons based on centerline junctions, with the following fundamental principle: (1) splitting the original road surface based on the dangling nodes in the road centerline [Figs. 1(1) and 1(2)], (2) initially extracting the boundary arc information associated with the current intersection according to the intersection nodes of the centerline [Fig. 1(3)], (3) calculating the information of the primary nearest point between the intersection nodes of the centerline and the associated boundary arc [Fig. 1(4)], (4) calculating the secondary nearest points of the associated boundary arcs except the edge where the primary nearest point is located [Fig. 1(5)], and (5) configuring the intersection surface sequentially for the secondary nearest points near each obtained intersection node [Fig. 1(6)], as shown in Fig. 1. The detailed process is shown in Fig. 2.

2.2 Extraction of dangling nodes

Build the topology on the basis of the original road surface and road centerline, extract all dangling nodes according to the road centerline topology information, and record the set of dangling nodes associated with the road surface rings, as shown in Fig. 3.

The specific steps for finding dangling nodes associated with the road line loop are as follows:

Step 1: Topologically preprocess road surfaces and centerlines to extract all dangling nodes.

Step 2: Turn the original road polygon into a line ring, build an R-tree spatial index, and add the line ring.

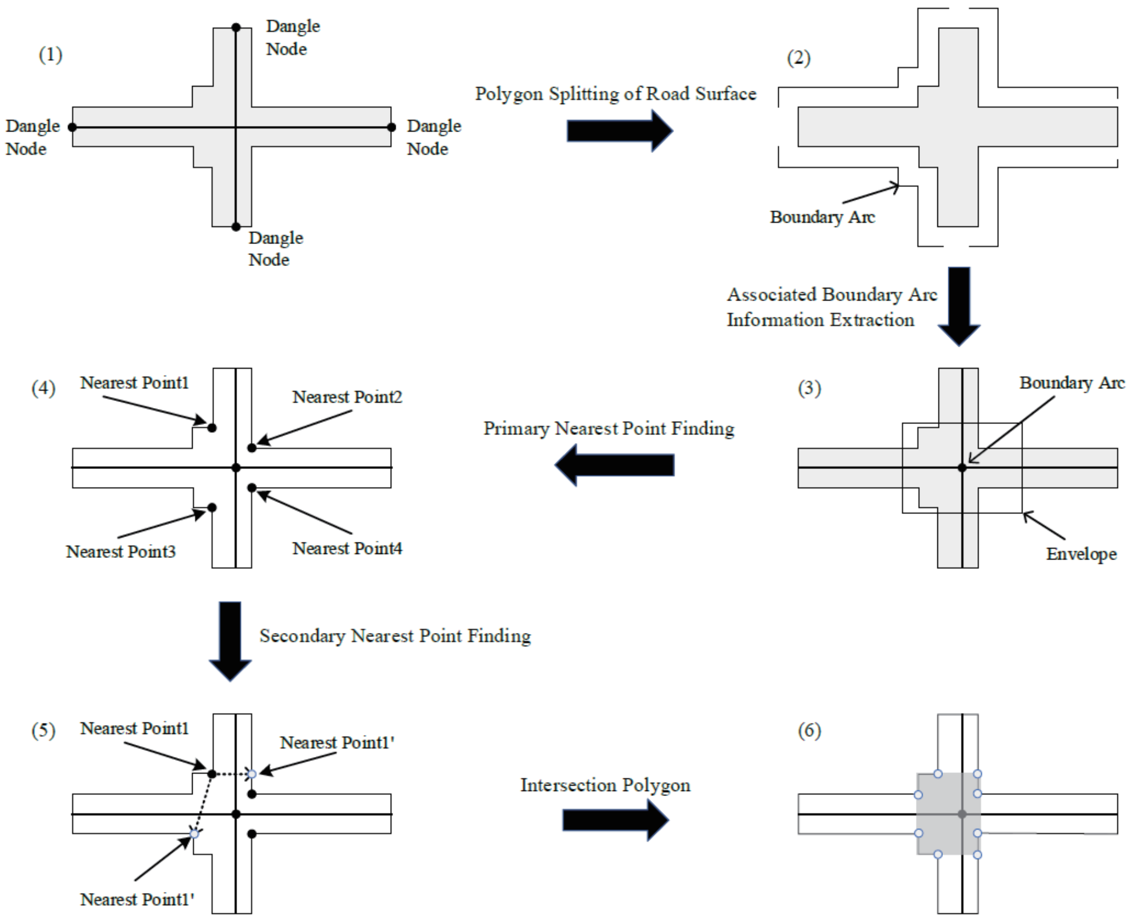


Fig. 1. Basic principle.

Step 3: Loop all dangling nodes, draw a rectangular box centered on each dangling node according to a certain threshold range and turn them into polygons, find the line ring intersecting the polygon space in the R-tree. If found, merge the dangling nodes and save them to the dictionary. Then, obtain the dictionary set of the line ring and the hanging point associated with the line ring after the loop is over and record it as γ.

2.3 Polygon splitting

The splitting of pavement polygons refers to the operation of breaking the original pavement polygons into multiple boundary arcs under the original pavement by using the dangling nodes associated with line rings. The purpose is to lay the groundwork for finding the nearest point later.

The basic approach to polygon splitting is as follows: first, the position information of all the dangling nodes associated with the wire ring is calculated, that is, the position index of the previous node, the position index of the dangling nodes, and the distance between the two nodes. To determine whether a new point needs to be inserted, the criterion is whether the current

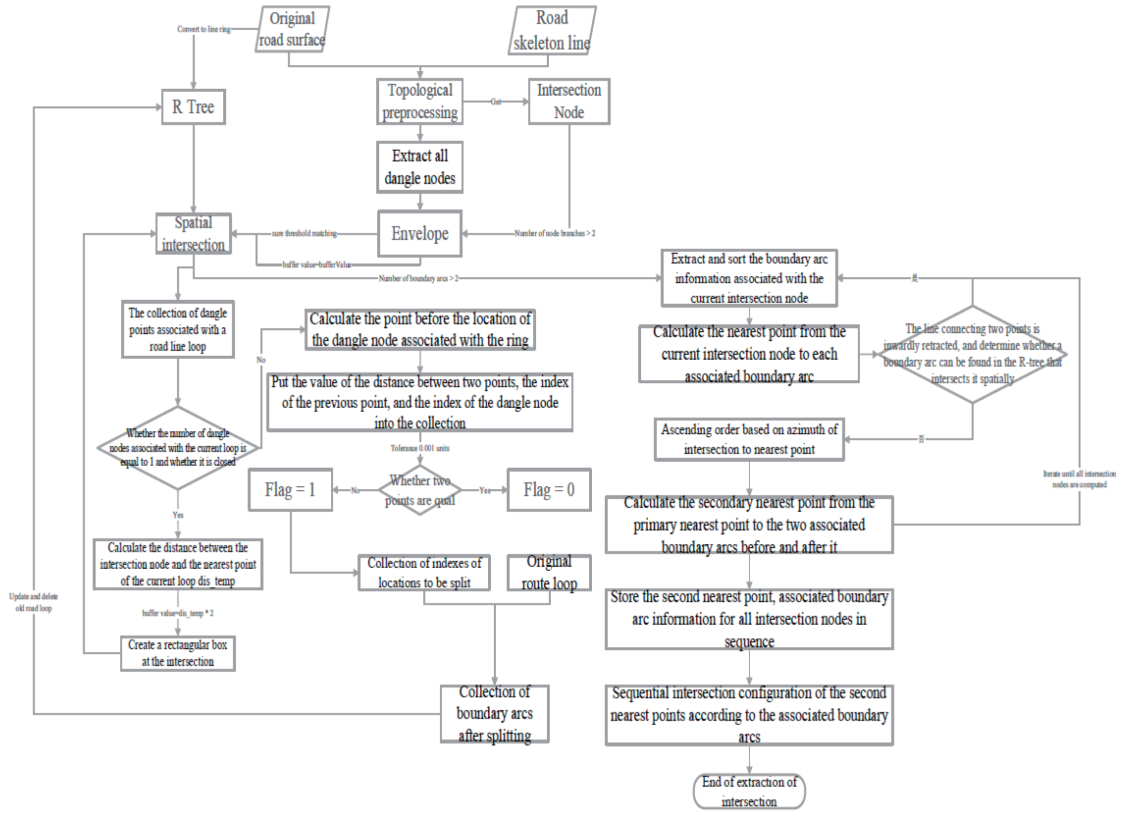


Fig. 2. Flowchart of extraction of intersection polygons based on centerline junctions.

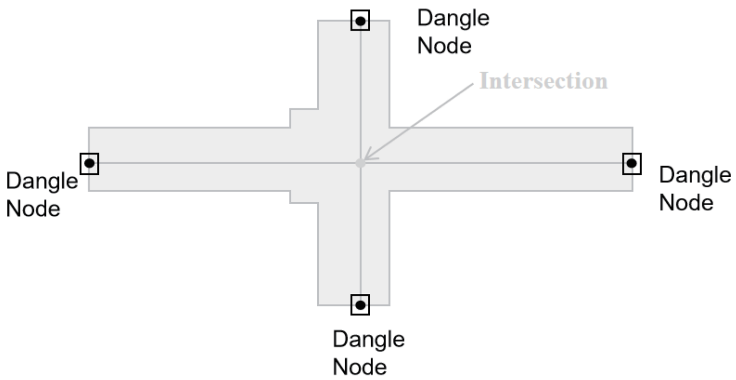


Fig. 3. Extraction of dangling nodes.

dangling node and the previous dangling node at its location are considered the same point within the set tolerance. Then, according to the mark, the split point is calculated on the basis of the position information of all dangling nodes associated with each line ring, and the split point set is obtained. The line ring is divided at the split point to obtain multiple boundary arcs, as shown in Fig. 4.

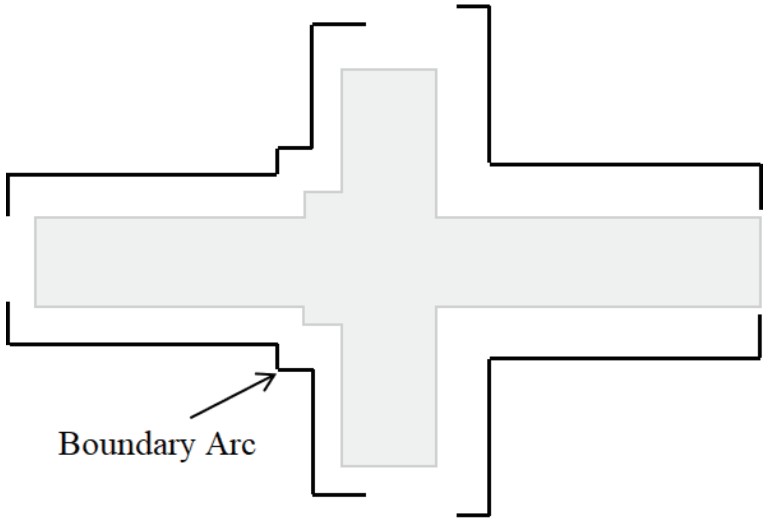


Fig. 4. Polygon splitting result.

The specific steps of polygon splitting are as follows:

Step 1: Before calculating the previous point where the dangling node associated with the central line ring is located, first, judge whether the number of dangling nodes associated with the current line ring is equal to 1 and whether it is closed, as shown in Fig. 5. If the conditions are met, calculate the distance D_{temp} between the intersection node and the nearest point of the current line ring and the two points. Take twice of this distance as the buffer threshold to build a rectangular box at the intersection, add the intersection point intersecting with the current line ring to the set of dangling nodes associated with the appropriate front ring, then calculate all the distances between all the suspended nodes associated with the updated line ring and the previous node at the location, and store all the calculated distances, the index of the previous point, and the index of the suspended node into the set, denoted as D_{vec}.

Step 2: Determine whether the distance between the previous node and the corresponding dangling node in the set D_{vec} obtained in Step 1 is less than the set tolerance. If yes, the two nodes are considered the same node. Here, the mark *Flag* is 0; otherwise, *Flag* is 1; finally, a set *Flags* composed of multiple marks is obtained.

Step 3: Create the index set of the location to be split, denoted as ψ. First, the set is added to the start position index of the loop. If the dangling node index corresponding to *Flag* equal to 1 in the set D_{vec} obtained in step 1 is not in the position index set ψ to be split, the index information is added to the set ψ until all the dangling nodes associated with the current loop are calculated. Finally, the end node index of the current loop is added to ψ.

Step 4: According to the index set of the position to be split, split the original route ring at each index position to obtain multiple boundary arcs that make up the road, add the R-tree, and delete the old route ring. Note that since the start and end of the line ring are added to the location index set to be split, the starting point index needs to be deleted. When there is no starting index in the location index set to be split, the boundary arcs under the

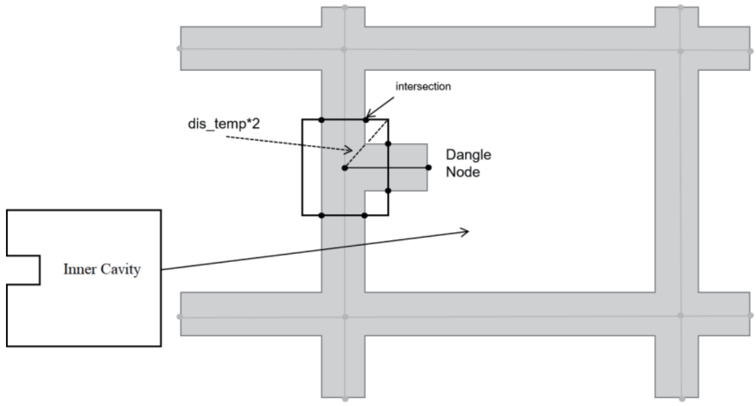


Fig. 5. Special case.

first and last indexes are spliced into new boundary arcs, and the location index set to be split is updated, that is, the I-th line ring split in the γ dictionary set is completed.

Step 5: Repeat steps 1 to 4 until the volume of the set is equal and the polygon split is complete.

2.4 Extraction of intersection polygon

The basic approach to the extraction of the intersection polygon is as follows: First, it is necessary to calculate the number of branches of intersection nodes as shown in Fig. 6 to ensure that the intersection surface has three or more forks. On the basis of the intersection node and associated boundary arc information obtained, the first closest point information is obtained. Finally, the information of the second closest point is obtained according to the information of the first closest point, and the intersection surface is formed by using the second closest point sequential plane.

The specific steps of intersection surface extraction are as follows:

Step 1: On the basis of topology preprocessing, the intersection nodes are obtained, and only the multi-fork nodes with more than three forks are stored, denoting them as ζ.

Step 2: Under the setting of the buffer threshold θ, where θ is the product of the initial threshold σ of the intersection identification buffer and the number of identification α, a rectangular box is established with the b-th intersection node of ζ, as shown in Fig. 7, and the boundary arcs intersecting the rectangle box space in the R-tree are searched. When the number of boundary arcs is greater than or equal to 3, a new set is stored and denoted as B. At this time, the boundary arc information of the current intersection association is extracted.

$$\theta = \sigma * 2^{\alpha}, \alpha = \left\{ \alpha \mid \alpha \in [0,10), \alpha \in N \right\}. \tag{1}$$

Step 3: Sort the extracted association boundary arcs of the current intersection and calculate the nearest point from the current intersection node to each association boundary arc. To prevent the occurrence of the inner ring, the current intersection node is temporarily connected to the calculated nearest point to form a line, and indentation is carried out

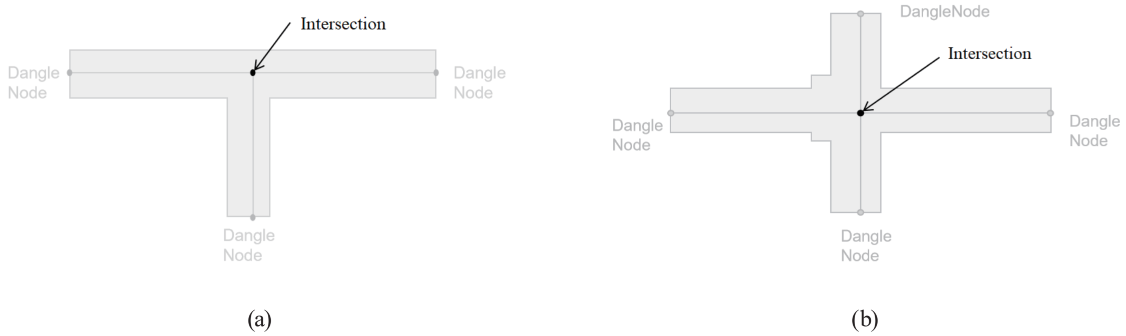


Fig. 6. Diagrams of intersection node. (a) Three-way intersection and (b) four-way intersection.

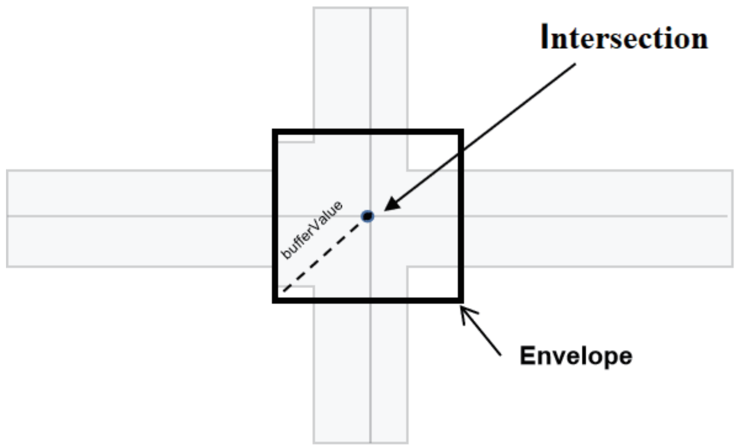


Fig. 7. Preliminary extraction of intersection-associated boundary arcs.

according to the set tolerance. If the boundary of the space intersecting with it can be found in the R-tree, the set obtained in step 2 needs to be updated, and the nearest point is discarded, and step 3 is carried out again.

Step 4: The collected V_1 is used to record the information of the first closest point of the current intersection and the associated boundary arc index obtained in step 3, and the information of the first closest point is calculated according to the ascending order of the azimuth angle between the intersection node and the nearest point, as shown in Fig. 8.

Step 5: According to the boundary arc associated with the first closest point, find the two associated boundary arcs before and after it, calculate the second closest point from the first closest point to the second closest point of the two associated boundary arcs before and after, as shown in Fig. 9, and store the information of the second closest point and the associated boundary arc of the current intersection node in sequence, denoting it as V_2.

Step 6: Repeat steps 2 to 5 until all intersections obtained in step 1 are calculated.

Step 7: Cycle the second nearest point at all intersection nodes and extract the intersection surface according to the associated boundary arc by order plane. The extraction of the intersection surface is completed, and the results are shown in Fig. 10.

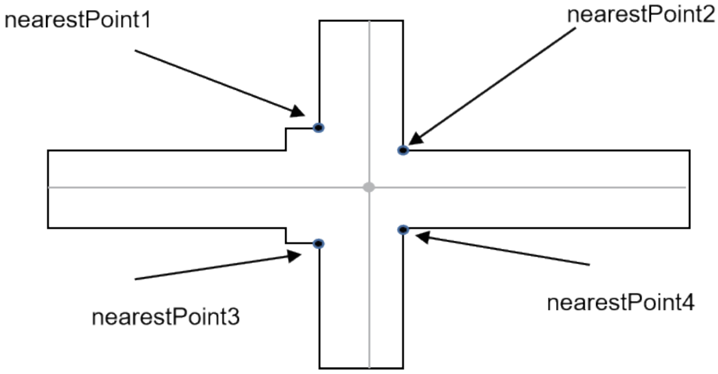


Fig. 8. Information of first nearest points.

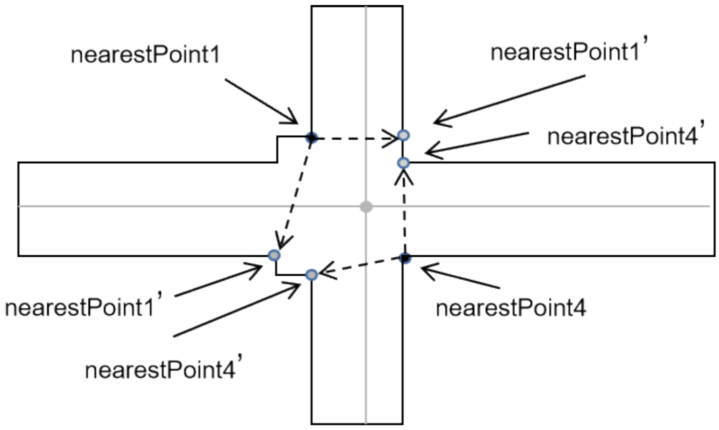


Fig. 9. Process of obtaining the second nearest points.

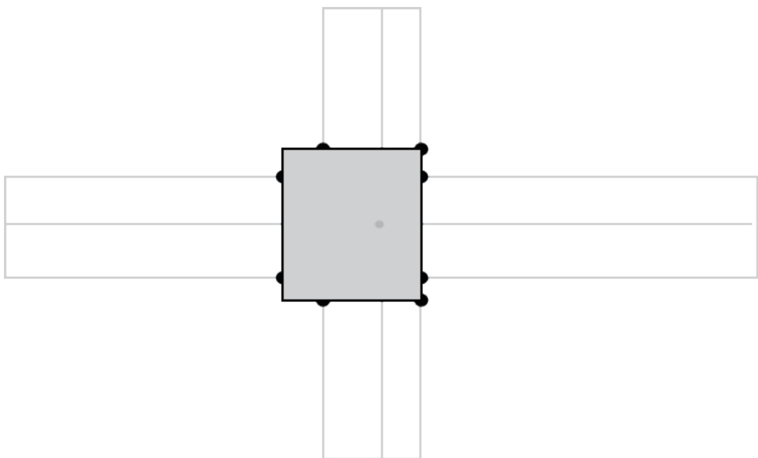


Fig. 10. Result of extraction of intersection polygon.

3. Experiment and Analysis

3.1 Experimental data and environment

To verify the effectiveness of the proposed method, relying on the WJ-III map workstation developed by the China Academy of Surveying and Mapping Science, an intersection surface extraction method based on the centerline bifurcation was embedded. The operation of the intersection surface extraction was carried out in a C++ environment. The experimental environment is a Microsoft Win7 64-bit operating system, the CPU is Intel Core I7-4790 (standalone 8 cores and 8 threads), the main frequency is 3.6 GHz, RAM is 8 GB, and the SSD (Solid State Drive) capacity is 1024 GB.

A series of road data with different scales covering a total area of 1337.143 km^2 were selected for the experiments. These areas included 205.353, 675.225, and 456.565 km^2. The regions are economically developed with dense road networks and comprehensive types of intersection. The total data volume was 165.51 MB in Shapefile format, including 137 source polygonal roads and 14,298 source linear roads. An experiment was conducted to compare between the proposed method and manual processing using the same dataset to validate the reliability of the proposed method.

3.2 Experimental analysis

The experimental parameters were set as follows: an intersection identification buffer threshold of 100 m, a node fitting threshold of 0.001 m, a line segment intersection tolerance of 0.00001 m, and a node redundancy difference of 0.001 m. On the basis of the proposed method, a total of 5196 intersections were extracted.

3.2.1 Visual evaluation

In the data processing stage, the proposed method utilizes dangling nodes obtained by topological processing to split the original road loops into multiple boundary arcs. These arcs are used to calculate the nearest point information from the intersection nodes to the boundary arcs, forming the required intersection areas. The algorithm is based on 2D vector data, effectively solving the extraction of planar intersections. Additionally, the method uses a second nearest point to other boundary arcs from the nearest point to form a more ideal intersection area (Fig. 11).

To better demonstrate the intersection extraction results, Fig. 12 shows a local magnification of the data. The intersection extraction results of the proposed method (blue areas) and manual extraction (red areas) are compared. The intersection types include Y-shaped intersections, T-shaped intersections, cross intersections, X-shaped intersections, roundabouts, and more complex misaligned intersections. Regardless of the intersection type, the proposed method effectively maintains data consistency and integrity from an overall perspective and considers its spatial relationship with road data, resulting in a good visual effect.

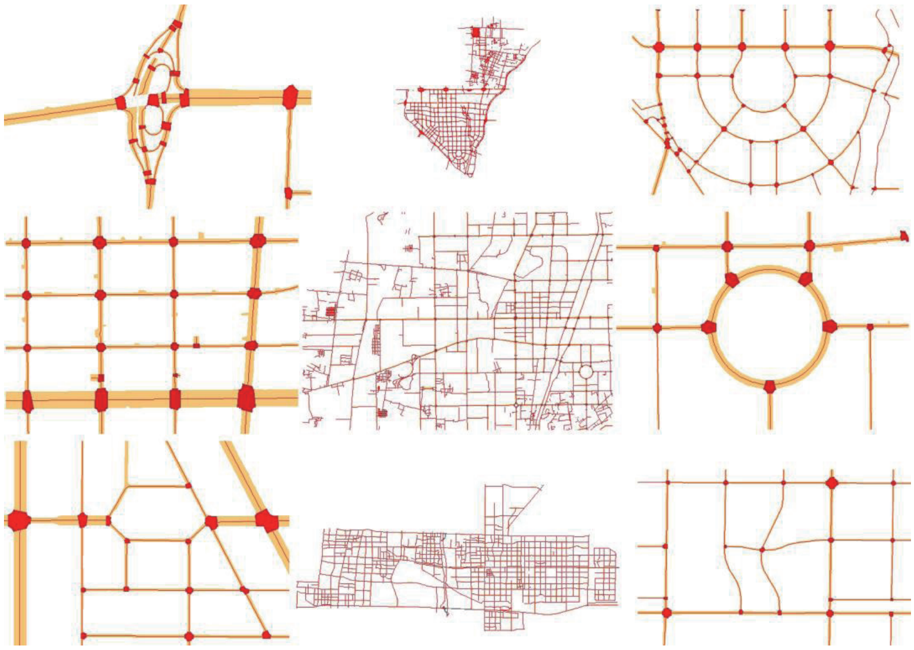


Fig. 11. (Color online) Experimental results.

On the basis of the visual cognitive effect of the above extraction results, the intersections extracted by the proposed method align well with the manually extracted results and can be processed automatically.

3.2.2 Statistical evaluation

To prove the high accuracy of the proposed method in extracting intersections, it was compared with the manual processing method. As shown in Table 1, the proposed method can accurately extract intersections.

On the basis of the proposed method, the accuracy of intersection extraction can reach 100%, indicating that the method can effectively extract intersections. Moreover, the accuracy of the proposed method meets application requirements and can better save manpower and material costs.

3.2.3 Efficiency evaluation

The total runtime of the proposed method was 422 s, with individual runtimes of 18, 386, and 18 s for the three trial data sets. However, the manual processing of the trial data required 1.5 days. In terms of runtime efficiency, the proposed method is more efficient and time-saving, with high recognition accuracy, iterative processing capabilities, and practicality for engineering applications.

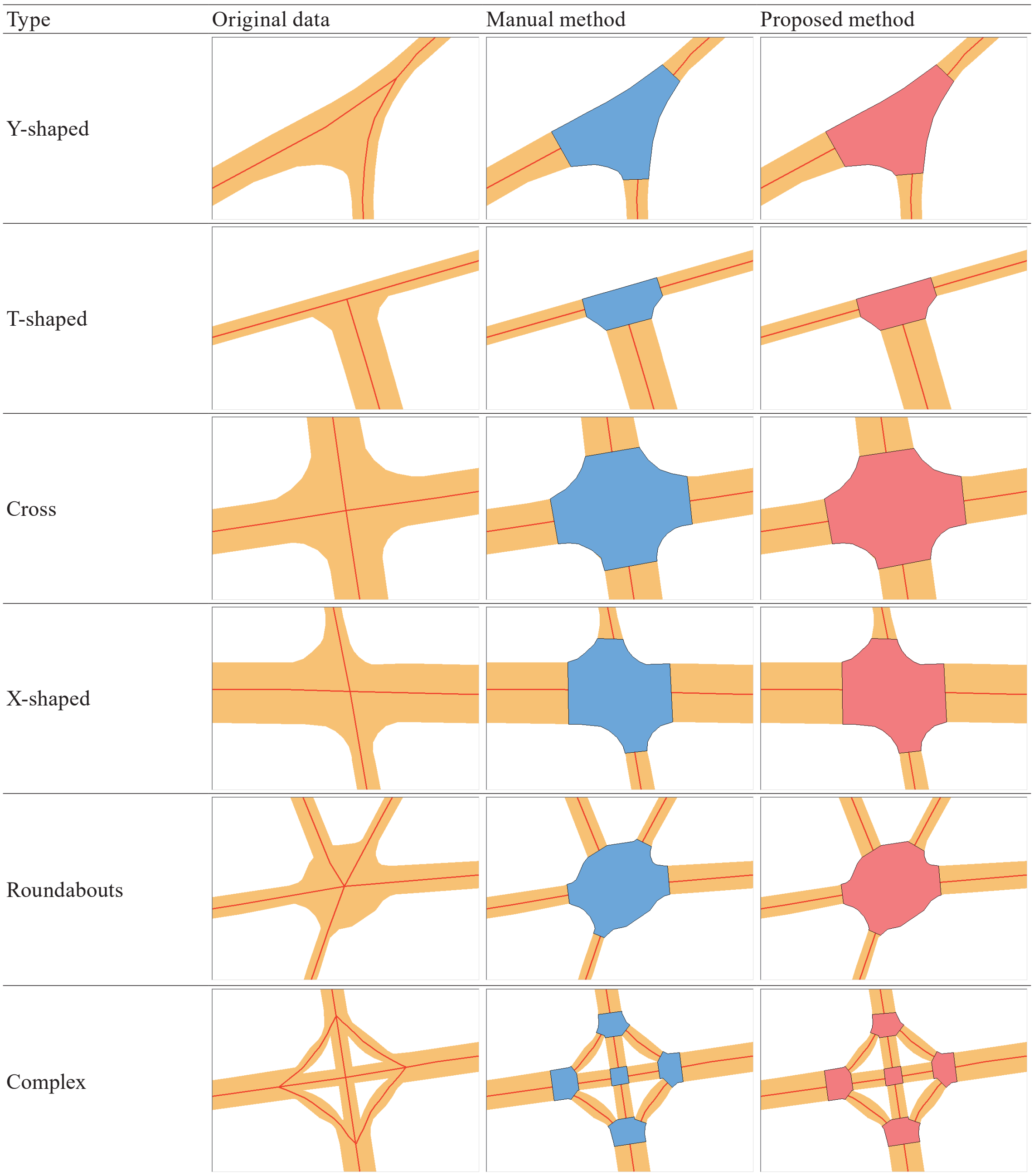


Fig. 12. (Color online) Comparison of intersection extraction results of two methods using local magnificatiion.

Table 1
Comparison table of two extraction methods.

Experimental data	Area 1	Area 2	Area 3
Original polygons/Count	26	107	4
Manual method/Count	786	3656	754
Proposed method/Count	786	3656	754
Accuracy rate/%	100	100	100

4. Conclusions

The experimental results showed that the proposed method based on the road centerline junctions can successfully extract the intersection polygons in the road network data and adapt to different intersection types. The research results can be used in application scenarios such as updating and enriching basic surveying and mapping data and extracting river surface intersections.

References

1 W. Zhang: Study on Design of Urban Road Intersection (2012).

2 L. Tang, L. Niu, X. Yang, X. Zhang, Q. Li, and S. Xiao: Acta Geod. et Cartographica Sin. **46** (2017) 770. https://doi.org/10.11947/j.AGCS.2017.20160614

3 S. Li, L. Xiang, C. Zhang, and J. J. Gong: Geo-Inf. Sci **21** (2019) 1845. https://doi.org/10.12082/dqxxkx.2019.190187

4 F. Wang, Q. Guo, and X. Xu: Sci. Surv. Mapp. **47** (2022) 212. https://doi.org/10.16251/j.cnki.1009-2307.2022.01.025

5 H. Hu, L. Xiang, and D. Wang: Bull. Surv. Mapp. **07** (2018) 53. https://doi.org/10.13474/j.cnki.11-2246.2018.0209

6 R. Li, Y. Si, D. Ma, and M. Qu: J. Geomatics Sci. Technol. **34** (2017) 168. https://doi.org/10.3969/j.issn.1673-6338.2017.02.011

7 G. Chen, L. Chen, G. Wang, X. He, and H. Sui: Sci. Surv. Mapp. **42** (2017) 148. https://doi.org/10.16251/j.cnki.1009-2307.2017.10.023

8 W. Zhou: Road Extraction from High Resolution Remote Sensing Images Based on Road Intersection. Master, 2018.

9 F. Guo, C. Li, Z. Li, H. Zhang, and Y. Liu: J. Geomatics Sci. Technol. **34** (2017) 199. https://doi.org/10.3969/j.issn.1673-6338.2017.02.016

10 Z. Chen, H. Ma, and Y. Li: Remote Sens. Land. Resour. **25** (2013) 79. https://doi.org/10.6046/gtzyyg.2013.04.13

11 J. Li, H. Liu, W. Xie, and X. Luo: Comput. Simul. **35** (2018) 161. https://doi.org/10.3969/j.issn.1006-9348.2018.11.033

12 H. Wen and Z. Tang: Highway Eng. **44** (2019) 55. https://doi.org/10.19782/j.cnki.1674-0610.2019.02.011

13 X. Zheng, Y. Chen, L. Jin, M. Zhang, and L. Xiang: China Saf. Sci. J. **28** (2018) 176. https://doi.org/10.16265/j.cnki.issn1003-3033.2018.10.029

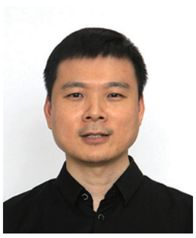
14 Q. Yu, R. Liang, Z. Guo, and X. Zhu: J. Bjing Univ. Technol. **45** (2019) 485. https://doi.org/10.11936/bjutxb2017120033

15 M. Chen and W. Wang: Geospatial Inf. **02** (2007) 14. https://doi.org/10.3969/j.issn.1672-4623.2007.02.006

16 C. Ma, Q. Sun, H. Chen, and Wen. B: Geomatics & Inf. Wuhan Univ. **41** (2016) 1232. https://doi.org/10.13203/j.whugis20160073

17 X. Ma, C. Li, and Z. Zhao: Acta Geod. et Cartographica Sin **43** (2014) 1083. https://doi.org/10.13485/j.cnki.11-2089.2014.0158

About the Authors



Lei Cai received his B.S. and Ph.D. degrees from Wuhan University, China, in 2003 and 2009, respectively. Since 2009, he has been an engineer at the Beijing Institute of Surveying and Mapping. He is the author of one book and more than 10 articles. His research interests include cartography generalization, toponymy, and historical and cultural geographic information systems and their applications. (28459050@qq.com)

Yong Yin received his M.S. degree from the Chinese Academy of Surveying and Mapping, Beijing, in 2008, and his Ph.D. degree from Shandong University of Science and Technology, Qingdao, in 2019. From 2017 to 2024, he was an associate research fellow at the Chinese Academy of Surveying and Mapping. He is the author of two books, and more than 15 SCI articles, and he has more than five inventions. His research interests include cartography generalization, computer visualization, geography, and geoinformatics. (yinyong@casm.ac.cn)

Qifei Mao was born in Taiyuan, Shanxi, China, in 2001. She received her B.Eng. degree in remote sensing from the Shandong University of Science and Technology, Qingdao, in 2023. She is pursuing a master's degree in cartography and geographical information engineering at Chinese Academy of Surveying and Mapping, Beijing.

Jiannan Yang is currently a Ph.D. candidate at Beijing Normal University, major in natural disaster science. He received his M.E. degree from the Chinese Academy of Surveying and Mapping, Beijing, in 2023, and has published 1 SCI article.