Data Modeling Method Based on CAD Intersection and Its Application

Xu Zhang, Min Huang, Jiajie Pan, and Jian Zheng

***Abstract*—Intersection is an important part of road network. In order to enrich the modeling methods of intersection, a data model of intersection is constructed, which contains *Link*, *Node*, *Arc*, *Lane*, and *LaneConnector*. With a planar CAD intersection map as input, geometric features and positional relationship between elements in the map are analyzed. Then the property data of the elements of the model, especially the logical topology data, are extracted, and a database of intersection model is constructed. Finally, a cross-shaped intersection is taken as an example to verify CAD intersection map's feasibility for constructing intersection model.**

***Index Terms*—CAD, data model, intersection, logical topology.**

I. Introduction

Road network is the basis of all research fields in intelligent transportation system. Traffic facilities intelligent manage, traffic information service system, and traffic simulation are in needs of the support of road network database. As an important part of the road network, intersection has many modeling methods, and its modeling data source is various. Some scholars [1]-[5] extract intersection data from aerial and remote sensing maps, with location, connected road segment, and road segment direction included. Some scholars [6]-[9] extract and update road network data based on massive GPS vehicle trajectory data. And in [10], it constructs an overpass data model with road vector data and digital surface model.

The road CAD intersection maps are abundant and readily available. In order to enrich the modeling methods of intersection, there is a need to construct an intersection data model and to extract data from road CAD intersection maps to construct an intersection database. In [11], calculating formulas of three types of lines which compose road centerline are given and the custom road curve can be drawn automatically based on ObjectARX. In [12], data of road segment, intersection, and road network are extracted from road CAD map successively to generate digital road map. Whereas, the most of the data got from these methods are physical topology data, with a lack of logical topology data.

Manuscript received October 8, 2014; revised February 15, 2015. This research was funded by the National Natural Science Foundation of China (No. 51178475, 11172333) and Science and Technology Program of Guangzhou, China (No. 2011J2200092).

The authors are with School of Engineering, Sun Yat-Sen University, Guangzhou, 510275 China and Guangdong Provincial Key Laboratory of Intelligent Transportation System, Guangzhou, 510006 China (e-mail: zhangx48@mail2.sysu.edu.cn, huangm7@mail.sysu.edu.cn, panjjie@mail2.sysu.edu.cn, zhengj52@mail2.sysu.edu.cn)

Based on lane-based road network model, this paper extracts data from road CAD intersection map to construct an intersection data model, which lays foundations for data modeling for road network based on road CAD engineering maps.

II. Intersection Data Model

With a rapid development of traffic, road network data model also experienced a considerable development. Among these models, lane-based road network model [13] can describe road network in multiple scales and from concrete to abstract perspective. It can also refine the road network to lane and describe the connectivity between lanes. The intersection data model constructed in this paper is based on the lane-based road network model.

A. Introduction to the Intersection Data Model

The intersection data model has elements such as *Link*, *Node*, *Arc*, *Lane*, and *Lane Connector*. This model can be divided into two levels from concept. The level-1 of the intersection data model is composed by the basic modeling unit such as *Link*, *Node*, and *Arc*. The elements described by the basic modeling unit in this level have the feature of constant traffic organization. Based on level-1 of the intersection data model, the level-2 of the intersection data model refines the road network to lane. It has elements such as *Lane* and *Lane Connector*. The connectivity between lanes is established in level-2. The lane-based intersection is shown in Fig. 1.

In Fig. 1, the intersection model includes the basic network modeling units. *Link* = {*Link*1, *Link*2, *Link*3, *Link*4}, *Link*1 = {*Arc*1, *Arc*2, *Arc*3}. The lane is defined on *Arc*, e.g., *Arc*1 = {*Lane*1, *Lane*2, *Lane*3}. *Lane Connector* can describe the connectivity between lanes, e.g., *LaneConnector*1 = {*Lane*3, *Lane*4}, which means that a vehicle can drive from lane 3 to lane 4.

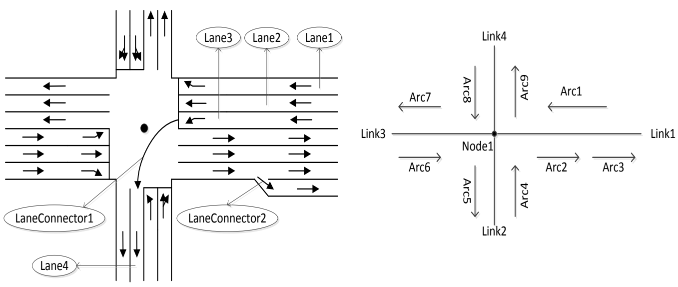


Fig. 1. Schematic diagram of lane-based intersection.

B. Structure of the Intersection Data Model

Based on the lane-based road network model, the Entity Relationship (E-R) Diagram of the constructed intersection data model is shown in Fig. 2. *Link* and *Node* compose the

DOI: 10.7763/IJMO.2015.V5.445

topology of the level-1 of the intersection data model. Link connects nodes. In the same link, the road's line shape, cross section, and traffic organization are constant. According to the actual traffic flow direction, a link can be separated into two different arcs, i.e., the topology direction of *Arc* is consistent with the traffic flow direction. The two arcs describe the topological connectivity of the actual roads. Attribute "Dir" of *Link* describes the one-way travel or two-way travel feature of the road. In a one-way road, "Dir" has the value of 1. In a two-way road, "Dir" has the value of 2. Attribute "AdjLink" of *Node* means the collection of links connected to the node. Attribute "LinkConnState" of *Node* tells the collection of connectivity between the links. Attribute "Dir" of *Arc* means its topology direction. It has the value of 1 when the topology direction is same with the traffic flow direction which enters the intersection. Otherwise, it has the value of -1.

LaneConnector is used to record the connectivity of each pair of lanes in an arc or different arcs. *LaneConnector* composes the topology of the level-2 of the intersection data model. It reflects topological connectivity in term of lanes. Attribute "Position" of *Lane* describes the number of the lane from left to right in the arc which contains the lane and its value starts from 1. Attribute "Change" of *Lane* reflects the change on broken and solid lines in the road. It has the values such as "Left", "Right", "Both", and "None". Attribute "TuringDir" of *LaneConnector* describes the turning rules at the end of the lane, e.g., "Straight", "Left", "Right", "U-turn", etc.

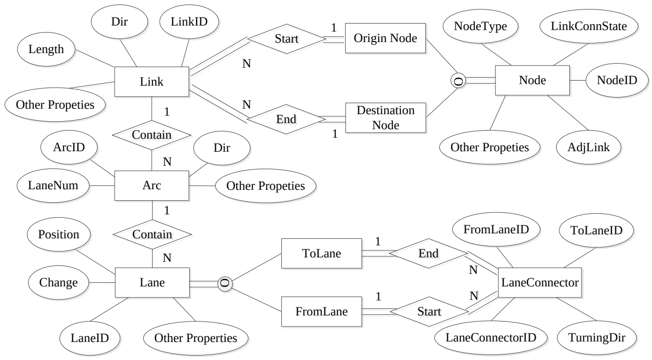


Fig. 2. E-R diagram of the intersection data model.

III. Modeling Data Extraction

The input for modeling data extraction is a CAD intersection map and the main input is the layers which contain various road markings and direction arrows. In order to extract and deduce the property data of the elements in the intersection data model, geometric features, physical property, and positional relationship between elements in the map should be analyzed. Then the intersection data model and database can be constructed.

A. Whole Progress

The whole process of modeling data extraction can be divided into five steps. The whole process of extracting modeling data is shown in Fig. 3.

1) Extract the roadway edge lines, roadway edge lines' centerline, and direction arrows. Whether the road is a one-way or two-way road can be determined by two parameters. The parameters are the distance between the roadway edge lines and positional relationship between the roadway edge lines' centerline and direction arrow. If the distance between the roadway edge lines equals to lane width, then the road is a road with only one lane. If the distance between the roadway edge lines is larger than lane width, and there is direction arrow on the left of the roadway edge lines' centerline, then the road is a road with multiple lanes. Otherwise, the road is a two-way road.
2) Generate *Link* according to the centerline of the roadway edge lines or the double yellow lines. Combined with the coordinates of the beginning and ending point of link, links will be numbered from 1 clockwise. Attribute "Dir" of *Link* can be gained from step one. Attribute "Length" of *Link* can be calculated by the coordinates of the beginning and ending point.
3) Connect facing links and use intersection point to generate *Node*. The links connected with the node can be recorded as the value of attribute "AdiLink" of *Node*. For an intersection, attribute "Type" has the default value of 1. The value of attribute "LinkConnState" of *Node* can be calculated by combining with discrimination of the direction arrow's type in the final step.
4) Extract roadway dividing line to construct *Lane* and *Arc*. Attribute "LaneNum" can be calculated by the distance between the roadway edge line and the centerline. Attribute "Position" of *Lane* can be calculated by the distance between the roadway dividing line and the centerline. For a two-way road, attribute "ArcID" of *Arc* can be determined by whether the Arc is on the left of the center line or on the right. The value of attribute "Change" of *Lane* can be calculated by combining with discrimination of the direction arrow's type in the final step.
5) Discriminate the direction arrow's type and deduce logical topology data. Each unique type of direction arrow has a unique direction arrow code. According to the direction arrow codes, the type of direction arrow can be discriminated. Then the connectivity between road and road and that between lane and lane can be deduced. The turning state of each lane can also be deduced. Finally, the property data of *LaneConnector* can be extracted.

B. Logical Topology Data Extraction

In an intersection map, direction arrows describe logical topology data of intersection, i.e., the connectivity between lanes. According to turning rules, the type of direction arrows can be classified into "Straight", "Left", "Right", "Straight and Left", *et al*. The connectivity data's deduction relies on the type of the direction arrows mostly, e.g., the right turning direction arrow means that there is connectivity between this lane and the outermost lane in the road on the right side. In a CAD map, a direction arrow is saved in the form of hatch. Its closed contour line is named as hatchloop. The discrimination process of direction arrows' type is shown in Fig. 4. With a straight direction arrow as an example, each element's definition in a direction arrow is shown in Fig. 5. Among it, the bulgevertex of a direction arrow means the turning points on direction arrow's hatchloop. The tip point means the point which faces the unclosed edge of the triangle part of a direction arrow. The bottom of a direction arrow means the shortest straight edge in the hatchloop except the straight lines which belong to the triangle part. At the same time, the edge on which the bottom lies is delimited as the down edge of the direction arrow's bounding rectangle.

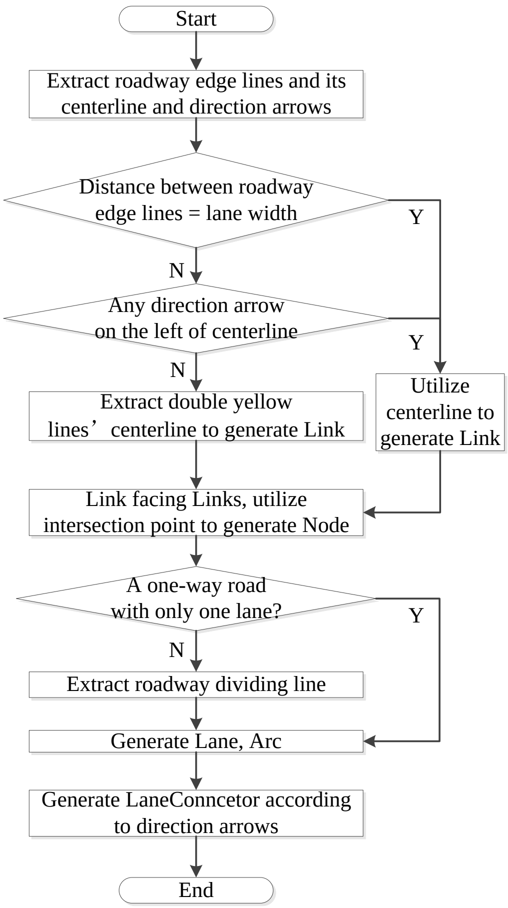


Fig. 3. Whole process of extracting modeling data.

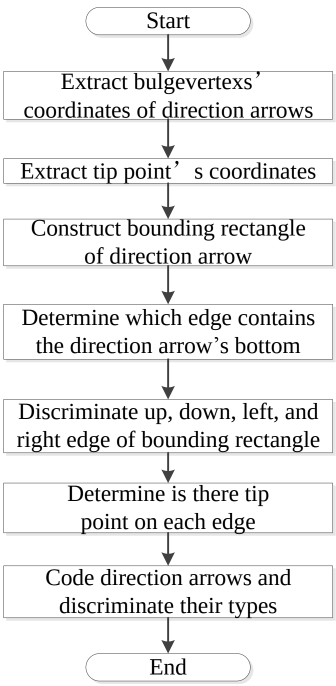


Fig. 4. Process of discriminating direction.

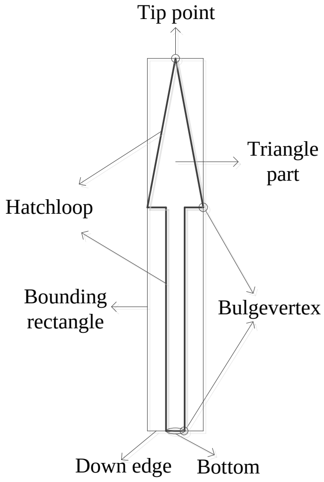


Fig. 5. Straight direction arrow.

According the sequence of up, down, left , and right, if there is a tip point on an edge of direction arrow's bounding rectangle, a code "1" is got, otherwise, "0" is got. In this case, a straight direction arrow has a code of "1000". Each direction arrow code corresponding to each direction arrow type is shown in Table I. Finally, the type of a direction arrow can be determined by the code, and then the connectivity data can be deduced by the direction arrow's type. Then, the property data of *LaneConnector*, such as "FromLaneID" and "ToLaneID", can be gained. By default rules, the laneconnector corresponding to right-turn arrow connects the lane on which arrow lies with the outermost lane in the road on the right side; the laneconnector corresponding to straight arrow connects the lane on which arrow lies with the corresponding lane in the facing road; the laneconnector corresponding to left-turn arrow connects the lane on which arrow lies with all lanes in the on the left side; the laneconnector corresponding to u-turn arrow connects the lane on which arrow lies with the innermost lane of the other travel direction in the same road.

TABLE I: DIRECTION ARROW CODE

Number	Direction arrow type	Direction arrow code
1	Straight	1000
2	Left-turn	0010
3	Right-turn	0001
4	Straight and left-turn	1010
5	Straight and right-turn	1001
6	U-turn	0000
7	Left-turn and U-turn	0110
8	Straight and U-turn	1100
9	Left-turn and right-turn	0011

IV. CASE STUDY

A typical cross-shaped intersection is selected as experiment area in this study. The CAD intersection map as input is shown in Fig. 6. The intersection established by the elements of intersection data model is shown in Fig. 7. The constructed intersection database contains tables such as *Link* table, *Node* table, *Arc* table, *Lane* table, and *LaneConnector* table as shown in Fig. 8. This database can meet application demand such as traffic simulation or traffic facilities manager at intersection smoothly. An application of constructing lane-based intersection in GIS is shown in Fig. 9.

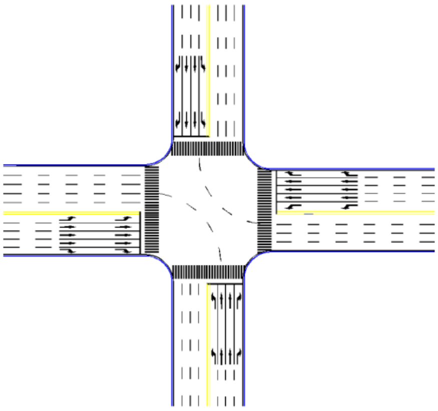


Fig. 6. CAD intersection map.

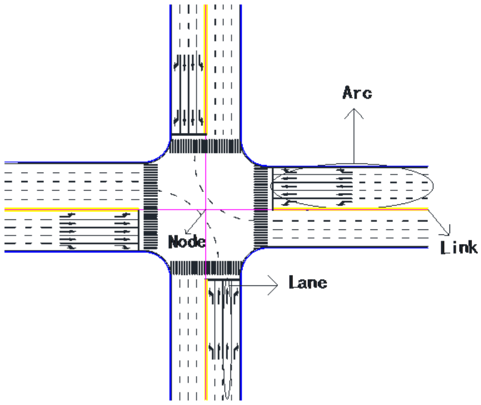
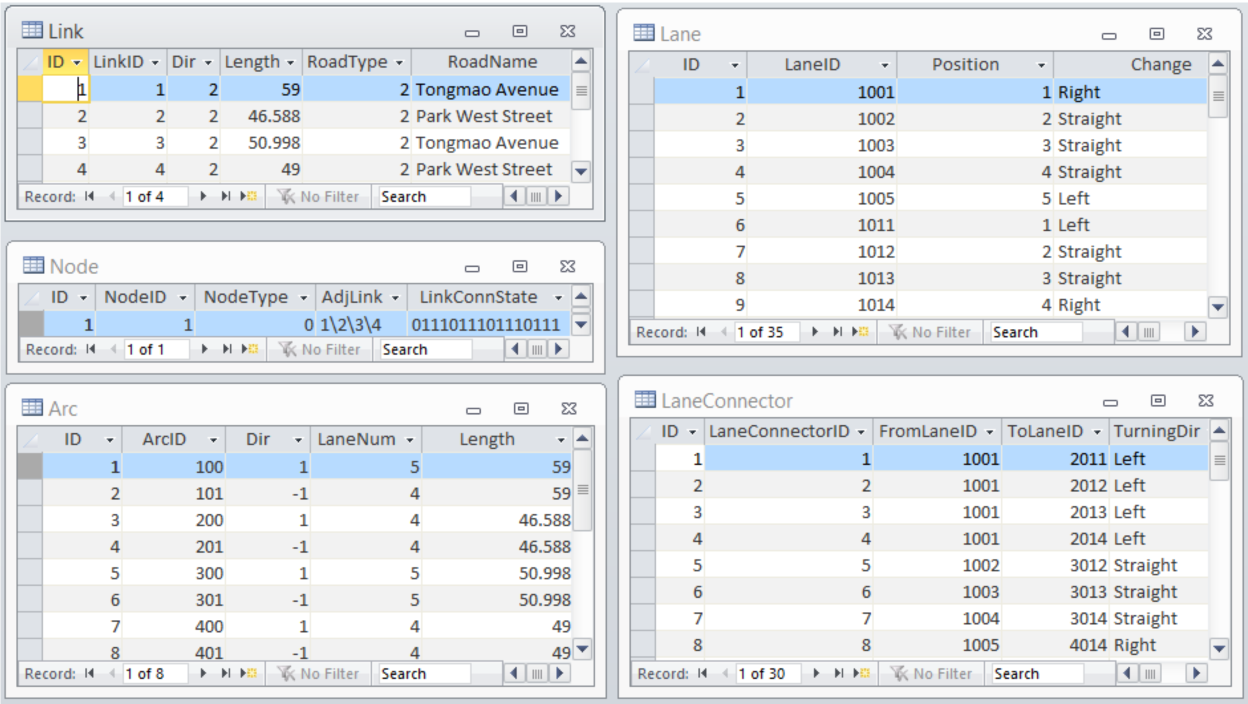


Fig. 7. Lane-based intersection map.



Link

ID	LinkID	Dir	Length	RoadType	RoadName
1	1	2	59	2	Tongmao Avenue
2	2	2	46.588	2	Park West Street
3	3	2	50.998	2	Tongmao Avenue
4	4	2	49	2	Park West Street

Record: 1 of 4 No Filter Search

Node

ID	NodeID	NodeType	AdjLink	LinkConnState
1	1	0	1\2\3\4	0111011101110111

Record: 1 of 1 No Filter Search

Arc

ID	ArcID	Dir	LaneNum	Length
1	100	1	5	59
2	101	-1	4	59
3	200	1	4	46.588
4	201	-1	4	46.588
5	300	1	5	50.998
6	301	-1	5	50.998
7	400	1	4	49
8	401	-1	4	49

Record: 1 of 8 No Filter Search

Lane

ID	LaneID	Position	Change
1	1001	1	Right
2	1002	2	Straight
3	1003	3	Straight
4	1004	4	Straight
5	1005	5	Left
6	1011	1	Left
7	1012	2	Straight
8	1013	3	Straight
9	1014	4	Right

Record: 1 of 35 No Filter Search

LaneConnector

ID	LaneConnectorID	FromLaneID	ToLaneID	TurningDir
1	1	1001	2011	Left
2	2	1001	2012	Left
3	3	1001	2013	Left
4	4	1001	2014	Left
5	5	1002	3012	Straight
6	6	1003	3013	Straight
7	7	1004	3014	Straight
8	8	1005	4014	Right

Record: 1 of 30 No Filter Search

Fig. 8. Intersection database.

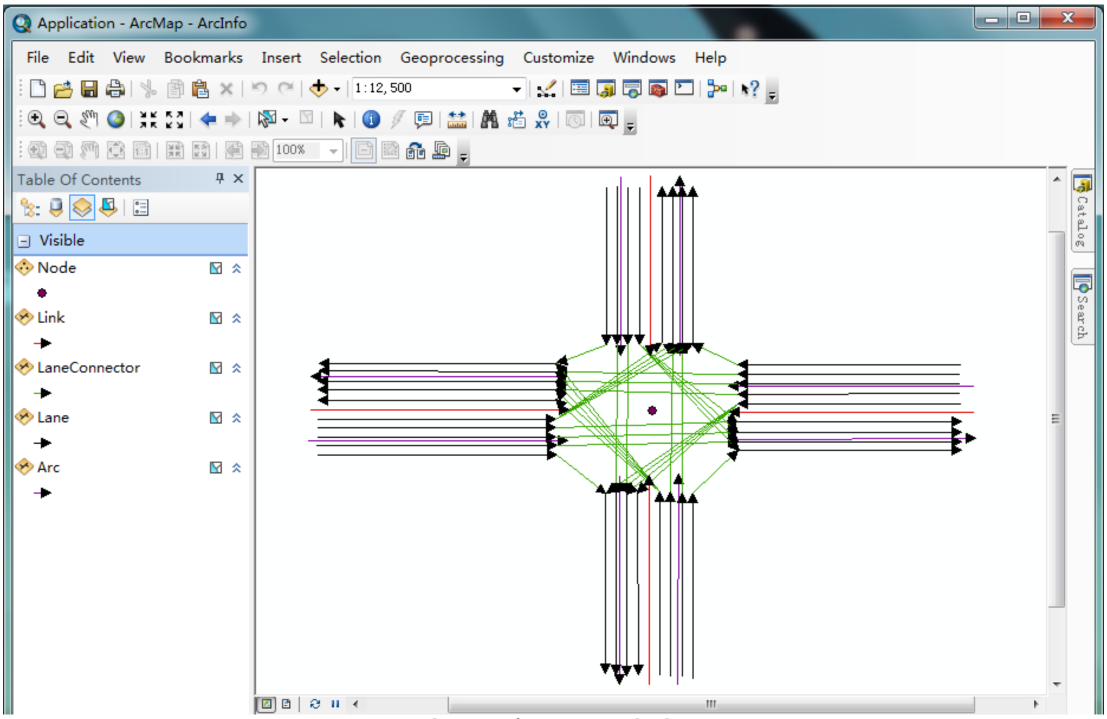


Fig. 9. Application of intersection database in GIS.

V. CONCLUSION

This paper proposes a method to construct an intersection data model and database based on CAD intersection map. By utilizing geometric features and positional relationship between elements in the map, the property data of each element of the intersection model are extracted successively to construct an intersection database. This method can enrich the modeling methods of intersection. It proves CAD intersection map's feasibility for extracting road network data to construct intersection model, and it lays foundations for data modeling for road network based on road CAD engineering maps.

ACKNOWLEDGMENT

The authors would like to thank the editor-in-chief and all the reviewers for their valuable comments and helpful suggestions to improve the quality of this article.

REFERENCES

[1] Y. Hu, K. Chehdi, G. Li, and K. Zu, "An optimal road seed extraction algorithm," in *Proc. the Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing(ETT and GRS 2008)*, December. 2008, pp. 771-775.

[2] Y. Y. Chiang, C. A. Knoblock, C. Shahabi, and C. C. Chen, "Automatic and accurate extraction of road intersections from raster maps," *GeoInformatica*, vol. 13, pp. 121-157, June 2009.

[3] M. Ravanbakhsh, C. Heipke, and K. Pakzad, "Road junction extraction from high-resolution aerial imagery," *The Photogrammetric Record*, vol. 23, pp. 405-423, Nov. 2008.

[4] K. Ho and P. Newman, "Multiple map intersection detection using visual appearance," in *Proc. the International Conference on Computational Intelligence, Robotics and Autonomous Systems*, December 2005.

[5] J. Hu, A. Razdan, J. C. Femiani, C. Ming, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," in *Proc. the IEEE Transactions on Geoscience and Remote Sensing*, 2007, vol. 45, pp. 4144-4157,

[6] W. Shi, S. Shen, and Y. Liu, "Automatic generation of road network map from massive GPS, vehicle trajectories," in *Proc. 12th International IEEE Conference on the Intelligent Transportation Systems, ITSC'09*, October 2009, pp. 1-6.

[7] L. L. Tang, F. Z. Huang, X. Y. Zhang, and X. Hong, "Road network change detection based on floating car data," *Journal of Networks*, vol. 7, pp. 1063-1070, July 2012.

[8] C. Chen and Y. Cheng, "Roads digital map generation with multi-track GPS data," in *Proc. the Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing(ETT and GRS 2008)*, December, 2008, pp. 508-511.

[9] L. Cao and J. Krumm, "From GPS traces to a routable road map," in *Proc. the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, pp. 3-12.

[10] J. Schpok, "Geometric overpass extraction from vector road data and dsms," in *Proc. the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2011, pp. 3-8.

[11] L. Yang and J. Zheng, "Custom road curve developed with objectARX," *Electronic Journal of Geotechnical Engineering*, vol. 18, pp. 1095-1108, March 2013.

[12] M. Chen and Y. Liu, "Recognition and extraction high precision digital road map," in *Proc. International Symposium on the Information Technology: Coding and Computing*, 2005, pp. 129-134.

[13] M. Huang, M. L. Rao, M. Li, and Z. M. Niu, "Lane-based road network model and its application in simulation road network modeling," in *Proc. the Applied Mechanics and Materials*, vol. 253, December 2013, pp. 1351-1355.

Xu Zhang was born in Lianyungang, China, in 1990. He received his B.S. degree in traffic engineering from Nanjing Forestry University, Nanjing, China, in 2013. He is currently working toward M.S. degree in School of Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include road network database construction.

Min Huang was born in Guangzhou, China, in 1975. She received her Ph.D. degree in engineering mechanics from Sun Yat-Sen University, Guangzhou, China, in 2006. She is currently a vice professor of School of Engineering of Sun Yat-Sen University. Her research interests include transportation network modeling and road guide sign system modeling.

Jiajie Pan was born in Qingyuan, China, in 1991. He received his B.S. degree in traffic engineering from Sun Yat-sen University, Guangzhou, China, in 2014. He is currently working toward M.S. degree in School of Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include road network database construction.

Jian Zheng was born in Cangzhou, China, in 1990. He received his B.S. degree in traffic engineering from Sun Yat-sen University, Guangzhou, China, in 2014. He is currently working toward M.S. degree in School of Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include road guide system.