COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2015; **26**:397–403
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1642

SPECIAL ISSUE PAPER

An efficient lane model for complex traffic simulation

Tianlu Mao[1]*, Hua Wang[2], Zhigang Deng[3] and Zhaoqi Wang[1]

[1] Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] School Of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China
[3] Computer Science Department, University of Houston, Houston, TX, USA

ABSTRACT

Traffic simulation heavily relies on lane model. This paper presents a novel method to model lanes based on the road axis under the Frenet frame. The road axis is generated from the geographic information system data after curve approximation, discretization, and compression. This lane model couples mileage information with three-dimensional geometric information, so it offers an easy and fast position transformation from mileage to the Cartesian coordinate. It also keeps strictly consistent for mileage among neighboring lanes so that it facilitates lane-change processing. Compared with existing methods that depict lanes as simple polylines or curves, the proposed lane model is more functional and more efficient, especially for complex traffic simulation with a large number of lane-changes. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS

lane model; the Frenet frame; traffic simulation; lane-changes

***Correspondence**

Tianlu Mao, No. 6 Ke Xue Yuan South Road, Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Hai Dian, Beijing, China.
E-mail: ltm@ict.ac.cn

1. INTRODUCTION

In a three-dimensional (3D) road network, traffic lanes are the basic elements, which describe the rightful travel way for vehicles. Lane data are queried and used when a traffic simulator inquires about the traffic state among neighboring lanes, plans lane-changing trajectories, calculates the motion states of vehicles along lane, and so on. Because of the high frequency of lane queries at every time step, how to efficiently model lanes has great influence on the effectiveness and efficiency of traffic simulation.

Existing traffic simulation systems [1–4] mainly depict lanes as polylines at a low level of details. Because the positions and directions of vehicles are calculated based on the lane data, the simulation results are visually unsmooth. Wilkie *et al.* developed a road model that transformed geographic information system (GIS) data into arcs with C^1 continuity [5]. Based on the arc-shaped lanes, they produced more accurate and smoother simulation results. However, it has several disadvantages. In traffic simulation, if a vehicle is moving forward along a lane, the simulator calculates the vehicle's one-dimensional (1D) motion state, velocity along the lane, and position offset along the lane. Furthermore, the position of a vehicle is described as a mileage number on a lane, which has to be transformed to the Cartesian coordinates with directional angle for 3D visualization. Wilkie's lanes model does not provide a direct and efficient mechanism to transform the vehicle's position from a mileage representation to its corresponding Cartesian coordinates. In addition, at the cases of lane-changing, a simulator has to query the traffic situation of neighboring lanes, make a lane-changing decision, and plan a lane-changing trajectory. In this regard, an efficient lane model is needed to facilitate fast neighboring queries and calculate positions in lane-changing simulation, which is fundamental to the efficiency of numerous traffic simulation systems.

To tackle the aforementioned challenges, in this paper, we present a novel efficient traffic lane model. It transforms GIS road data into spline curves and generates lane data under the Frenet frame that is locally orthonormal and in C^2 continuity. The spline curves are parameterized by arc-length. Our model naturally blends 3D geographic information with 1D mileage information. Therefore, it could easily facilitate a fast 3D position transform from a mileage representation to its corresponding Cartesian coordinates. Additionally, our model keeps strict consistency of mileage information between neighboring lanes, so it can enable traffic simulators to perform fast queries on parallel positions on neighboring lanes and facilitate realistic lane-changing simulation.

We further optimize the lane model by discretizing spline curves to polylines and compressing them based on user specification. Independent of any particular traffic simulator, our model is general such that it can be straightforwardly plugged into any existing traffic simulation systems. Our experiments also demonstrate how our lane model can be efficiently used for position transform, parallel neighboring query, and lane-changing.

The main contribution of the presented lane model is that it couples two types of lane information together: geometric information and locomotion-related information. The latter includes mileage and direction angle of the tangent vector. With such a hybrid structure, it facilitates fast position transform and realistic lane-changing process and, consequently, facilitates highly efficient traffic simulation.

2. RELATED WORK

2.1. Traffic Lane Models

How to depict traffic lanes is a fundamental yet important problem in traffic simulations. Most of existing traffic simulation methodologies and systems depict lanes as polylines [2–4,6–9]. Yang *et al.* presented a method for road network descriptions with polyline lanes [10]. This method is used in the traffic simulation software MITSIM [1]. In these methods, polylines of lanes come from the GIS data. However, if the input geographic data lack details, the simulation results are visually unsmooth.

Wilkie *et al.* introduced a new approach to automatically transform GIS data into functional road models [5]. They presented a formula to represent roads as arcs and line segments with C^1 continuity, which is closer to real roads. By offsetting the arc road, they generated arc lanes and produced smoother traffic simulation results. However, because the arc lanes, are unrelated to the mileage information, the Cartesian coordinates of vehicles cannot be quickly and directly transformed from the mileage position for 3D visualization. Additional computing is needed for such position transform.

2.2. Moving Frames

The surface of road constructed by multiple traffic lanes is a curved surface. Usually, it is generated by extending road axis curves. It is easier to depict curved surfaces in a moving frame than in the Cartesian system. There are two major moving frames for curved surfaces: the Frenet frame [11–13] and the rotation-minimizing frame [11,14,15].

The Frenet frame is a well-known approach in the tracking control theory and has been widely used in the geometric modeling community. It describes the geometric properties of a curve with C^2 continuous. In the frame, the unit vector is tangent to the curve; the normal unit vector and the binormal unit vector together form a local right-handed rectangular coordinate system. The Frenet frame has the invariance of movements, and every parameter in the frame has its specific geometric meaning. Xu *et al.* used this frame to generate vehicles' trajectories in traffic simulations [16]. However, they did not give a lane model under the frame. The rotation-minimizing frame is an attractive frame to model sweep surface. It has also been widely used in computer animations.

In this work, we model traffic lanes under the Frenet frame, naturally blend 3D geometric and mileage information of lanes, and couple mileage information among neighbor lanes to meet the requirement of fast and complex traffic simulation.

3. CONTINUOUS LANE MODEL

In this section, we will first shortly introduce the Frenet frame. Then, in order to model traffic lanes with C^2 continuity in the Frenet frame, we transform GIS road data into arc-length parameterized curves. According to the locally orthonormal characteristic of the Frenet frame, we give a continuous representation of our lane model. In the next section, we will discretize and compress the continuous ones to further optimize our lane model.

3.1. The Frenet Frame

The Frenet frame is a moving reference frame to describe the geometric properties of a continuous curve. It is an important frame to depict the local properties of the curve in the 3D Euclidean space [17].

The arc-length smooth parametric curve is

$$P(s) = (x(s), y(s), z(s))$$

The term $(x(s), y(s), z(s))$ is the Cartesian coordinate of the point where s is the arc-length parameter. Assume t is obtained as the derivative of $P(s), t = t(s) = P' = \frac{dP}{ds}$. Given $t = \frac{t}{|t|}$, then

$$t^2 = t \cdot t = 1$$

Assuming vector $n, n = n(s) = \frac{t'}{|t'|}$, called the primary normal vector of a curve at the point $P(s)$. Then, the third vector is $b : b = t \times n$. These three vectors, $t, n,$ and b, are orthogonal to each other and compose an orthonormal right frame attached to the point $P(s)$. This frame is named the Frenet frame. It is a moving frame: these three vectors, $t, n,$ and b, of the frame change while moving along the curve [18].

In the Frenet frame, t is defined as the derivative of $P(s)$ and n is the second derivative of $P(s)$, which means the first and second derivatives of any point on $P(s)$ should exist. So, $P(s)$ is C^2 continuous.

3.2. Curve Approximation for a Road Axis

The Frenet frame is a local coordinate system related to the arc-length parameter. In the traffic simulation, the motion state of each vehicle, when traveling along the lane, could

Comp. Anim. Virtual Worlds 2015; **26**:397–403 © 2015 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

be naturally depicted by its velocity along the lane and the position offset along the lane. Thus, constructing a lane model under the Frenet frame provides direct transformation from a 1D position along a lane (mileage) to a 3D Euclidean space position.

When modeling lanes for a traffic simulation, one of the common inputs is polylines of road axis, which could be downloaded from the online GIS system, like GoogleMap. In this paper, it generates the lane data under the Frenet frame, which is constructed on an arc-length parameterized curve. So, the method of generating arc-length parameterized curve for road axis in polylines needs to be first discussed.

A cubic spline is a piecewise cubic function: its first and second derivatives are both continuous. In this paper, we applied a technique pointed out by Wang *et al.* [19] to generate arc-length parameterized cubic spline curves to represent a road axis as

$$P(s) = \{P_1(s), P_2(s), \ldots, P_n(s)\}$$

$P(s)$ is a set of piecewise cubic functions, and $P_i(s) = (x(s), y(s), z(s))(i \in [1, n])$ is a piecewise cubic function with its arc-length parameter s.

There are linear segments among the curves of a road axis. So, some of the functions in $P(s)$ are not cubic but linear, which only have either first derivatives or no derivative.

3.3. Lanes

Lanes are the curves with a certain distance from road axis, which cannot be generated by panning the road axis directly, especially in the curved part. However, defining the road axis in the Frenet frame supports generating lanes from the road axis. This work gives the modeling method of traffic lanes in the two-dimensional space.

Assume a point $(x, y) = P_i(s)(P_i(s) \in P(s))$ is the original point on the road axis (as shown in Figure 1). According to Section 3.1, the vector $t(s) = P'(s)$ and $n(s) = t'(s)$ compose the two-dimensional Frenet frame. In fact, $t(s)$ and $n(s)$ are the vectors, respectively, in the tangent and normal directions at the point (x, y). Therefore, every point (x, y) on a road axis could be represented as $(s, 0)$ in the Frenet frame, where s is the arc-length parameter of the point (x, y). If the distance between the road axis and the lane axis of this road is l, then every point on the lane axis could be represented as (s, l).

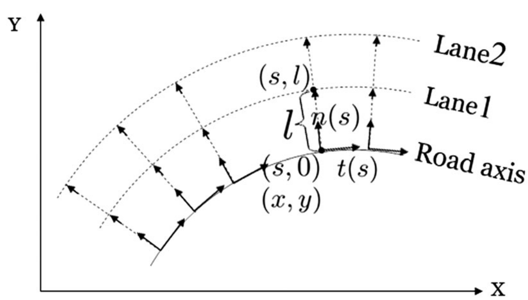


Figure 1. Generating lane curves from road axis.

According to road axis and l, the position of each point could be represented by the Cartesian coordinate of (s, l) in traffic simulations. The representation of lane is

$$\tilde{P}(s,l) = \left\{\tilde{P_1}(s,l), \tilde{P_2}(s,l), \ldots, \tilde{P_n}(s,l)\right\} \tag{1}$$

Here,

$$\tilde{P}_i(s,l) = \tilde{P}_i(s,0) + \begin{pmatrix} -l\sin\theta \\ l\cos\theta \end{pmatrix} \tag{2}$$

And,

$$\tilde{P}_i(s,0) = P_i(s) \tag{3}$$

θ is the direction angle of the tangent vector at the point $(s, 0)$ along the road axis curve. It is possible to obtain multi-lanes of a road by giving different l.

Assume the Cartesian coordinate of (s, l) is (x^1, y^1), then according to Equations (1)–(3),

$$(x^1, y^1) = \tilde{P}_i(s,l) = \begin{pmatrix} x - l\sin\theta \\ y + l\cos\theta \end{pmatrix} \tag{4}$$

It is remarkable that the Frenet frame works only when the curve is C^2 continuous. However, there are straight segments in road axis curves, where $n(s)$ is indeterminate in the Frenet frame. In such cases, $n(s)$ is calculated by the following formula based on the fundamental of the Frenet frame:

$$n(s) = t(s)\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Here, $n(s)$ is perpendicular to $t(s)$, and these two vectors can compose a right frame.

4. DISCRETIZED LANE MODEL

The continuous lane model depicts lanes as curves in the Frenet frame. It can give a visual smooth appearance in traffic simulations. It also gives an explicit transformation from a 1D position along the lane (mileage) to a 3D Euclidean space position. Although such transformation is fast, it still consumes certain computational resources, because the curve formula needs further processing. Thus, this paper presents a discretized lane model to simplify and fasten the transformation.

4.1. Compressive Discretization for Road Axis Curve

Because lanes are generated from the road axis, the road axis is firstly discretized. The Douglas–Peucker algorithm [20] has been applied in this work, which is a compressive discretization algorithm aimed to approximate a curve by a set of points through a compression threshold ε. The max distance should be smaller than ε between a curve

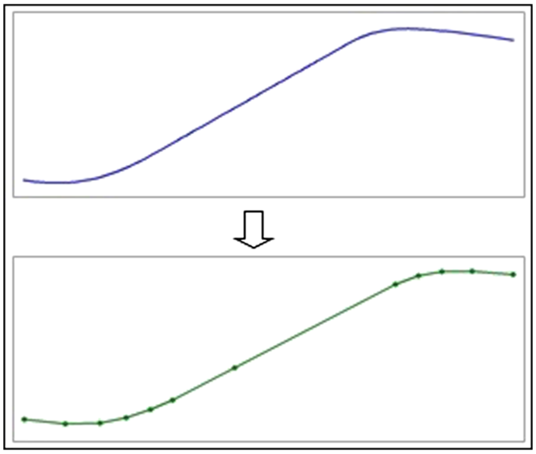


Figure 2. A compression example of a curve.

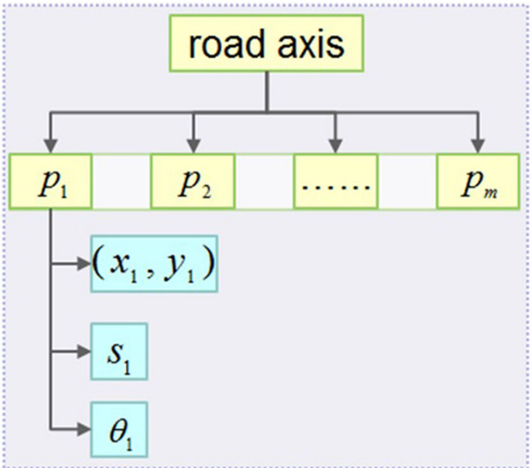


Figure 3. The data structure of a road axis.

and the polyline composed by the points. Figure 2 shows the compression result of a curve. The top panel shows the original curve, and the bottom one shows its corresponding the compression polyline. The following point set describes the road axis:

$$p = \{p_1, p_2, \ldots, p_m\} \tag{5}$$

The number of points for p is determined by ε. A smaller threshold ε would lead to a larger number of points. Meanwhile, the polyline composed by these points will be much closer to the original curve.

We store the angle of the tangent vector and the mileage of each point. So, the coordinate of a vehicle only depends on a linear interpolation. The data structure for a road axis is shown in Figure 3. The term (x_1, y_1) is the Cartesian coordinate; s_1 is the mileage; and θ_1 is the direction angle of the tangent vector for p_1.

4.2. Lane Representation

In order to model the lanes using the discretized road axis, θ in Equation (4) is defined as follows (assume the Cartesian coordinates of a given point on road axis is (x, y)):

(1) If $(x, y) \equiv p_i, \theta$ is known because we have stored it in the process of the compressive discretization for the curve of the road axis.
(2) Otherwise, if (x, y) is between p_{i-1} and p_i, θ is equal to the direction angle of the vector $\overrightarrow{p_i p_{i+1}}$.

In this way, although we depict road axis by polylines, we can still depict lanes through the method described in Section 3.3. In sum, a lane is represented by road axis, a set of points, an offset, and the distance between the lane and its road axis (Figure 4).

In this discretized lane model, the offsets are input data. The terms $(x_1, y_1), s_1$, and θ_1 are obtained by the discretization of the curves for road axes.

5. SELECTED APPLICATIONS

5.1. Fast Position Transform

In traffic simulations, if a vehicle moves forward along Lane2 (Figure 5), our approach can then determine its Cartesian coordinate based on Equation (4). In this

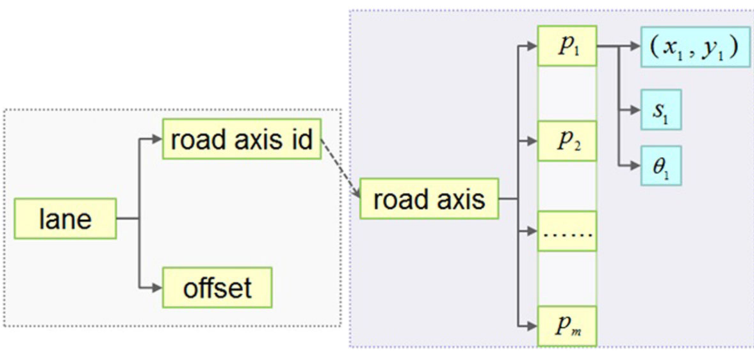


Figure 4. The data structure of a lane.

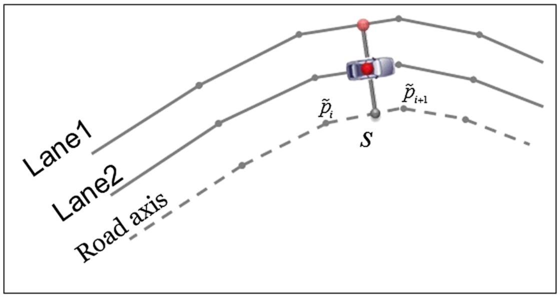


Figure 5. An illustration of position determination on a lane.

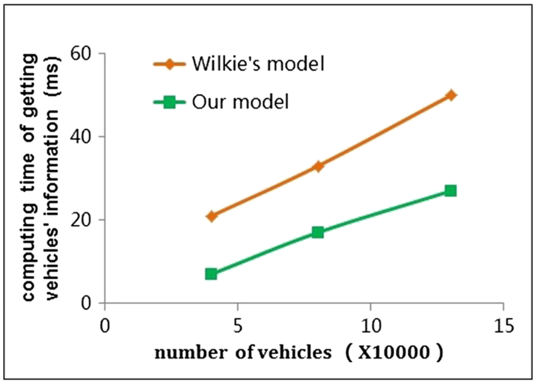


Figure 6. Efficiency comparison of position determination on a single lane between our model and the Wilkie's model [5].

equation, every parameter is known except the Cartesian coordinates (x, y) of the parallel position on the road axis. In fact, it only needs a linear interpolation to obtain (x, y) using our data structure of road axis,

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{s_{i+1} - s_i}\left[(s - s_i)\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} + (s_{i+1} - s)\begin{pmatrix} x_i \\ y_i \end{pmatrix}\right]$$

The term s is the mileage of which the vehicle locates; s_i is the mileage of p_i and $s_i \le s < s_{i+1}$; and (x_i, y_i) is the Cartesian coordinates of p_i.

We have compared the efficiency of position transform between our model and Wilkie's method [5]. These results were collected on an Intel Xeon E31240 processor running at 3.3 GHz. As shown in Figure 6, our method is about two times faster than the Wilkies's method. The absolute gap value of the processing time, only about tens of milliseconds, is not large. However, in a 3D traffic simulation, the simulation frame value stays about 50 milliseconds. Painting and rendering are also needed.

5.2. Facilitate Lane-changing

If vehicle changes its lane to the target lane (Lane1 in Figure 5), it has to query the traffic information of the Lane1 near its parallel position. In our model, the mileages of the parallel positions of these two neighboring lanes are the same. The Cartesian coordinate of the parallel position on the target lane could also be obtained from Equation (4). The only difference is the offset.

Our model keeps strict consistency of mileage information between neighbor lanes, so it could quickly compute a parallel position location for lane-changes. If a lane model depicts lanes with curves or polylines individually, the accumulated arc-length or polyline length of the parallel positions between neighboring lanes is different because lanes are not always straight, it needs to locate parallel positions on the target lane through a minimum distance algorithm or an intersecting operation, which is obviously more time consuming.

In lane-change planning, the trajectory of a vehicle could be easily described using the Frenet frame. By using this model, we could decompose a vehicle's movement into lateral and longitudinal displacements along lane curves. Obviously, $\vec{t}$ describes the longitudinal displacement, and $\vec{n}$ describes the lateral displacement in the Frenet frame. It defines the road axis point where a vehicle starts to change its lane, which corresponds to the origin of the Frenet frame. Then, the longitudinal and lateral displacements of the lane-change vehicle are $\phi(\tau)$ and $\varphi(\tau)$, which are functions of time τ. Applying polynomial equations to describe these two functions [16,21], then

$$\begin{cases} \phi(\tau) = A_5\tau^5 + A_4\tau^4 + A_3\tau^3 + A_2\tau^2 + A_1\tau + A_0 \\ \varphi(\tau) = B_5\tau^5 + B_4\tau^4 + B_3\tau^3 + B_2\tau^2 + B_1\tau + B_0 \end{cases}$$

Here, A_i and $B_i (i \in [0, 5])$ are the coefficients of the polynomial.

With our model, if a vehicle does not change lanes, $\varphi(\tau) \equiv 0$. Otherwise, $\varphi(\tau)$ increases from 0 to the value of the width between the current lane and the target lane in the whole lane-change process. After the lane-change planning, the Cartesian coordinate of the vehicle could be fast calculated by the longitudinal and lateral displacements under the Frenet frame.

In brief, two simple polynomials as functions of time could represent vehicles' complicated lane-changing movements. It is also straightforward to combine the vehicles' moving forward movements and lane-changing.

5.3. Storage

The geographic information of lanes in our method is depicted by road axis polylines and some offsets. In this case, it only needs to be stored in our database including those road axis polylines and offsets. Although the amount of storage increases as road networks become larger, the amount of storage for a road geographic data is nearly unchanged no matter how many lanes the road has and no matter what directions they are along (one-way or two-way).

This work allows the Douglas–Peucker compression algorithm to generate road axis polylines. The change of the compression threshold value in the algorithm will affect the storage amount. The amount becomes larger, and the polylines become much smoother, when the value becomes smaller. This means the user could specify the smoothness and the storage of our lane model according to specific applications.

6. CONCLUSION

We present a novel traffic lane model to facilitate highly efficient traffic simulation. According to the principle of the Frenet frame, we depict lanes by road axis and lane offset, regardless how many lanes the road has. The road axis is polylines coupling with mileage information and direction angle of the tangent vector. The polylines are generated from the raw data after curve approximation, discretization, and compression. The major advantages of our method are as follows: first, it is efficient to transform a mileage position to the Cartesian coordinate of a point on the lane. Second, it facilitates the lane-changing process because the mileage information among neighboring lanes in this model keeps consistent. Last but not the least, the lanes are visually smooth, even though we only store one polyline and some offsets for a two-way road.

Our current approach also has certain limitations. The vehicles might move on outer lanes or inner lanes, but not on the road axis. The mileage of the road axis has a small flaw with the length of different lanes. However, we regard them as they are the same. Fortunately, its impact on the movement of the vehicle is in the allowable range. We expect that our model can be straightforwardly incorporated into many existing traffic simulators; in the future, we would like to further develop an accurate road network model based on this lane model.

ACKNOWLEDGEMENTS

This work is supported and funded by the National Natural Science Foundation of China (grant nos. 61402269, 61173053, and 61272322) and the National Key Technology R and D Program of China (grant no. 2013AA013902). Z. D. was supported by the Open Project Program of the State Key Lab of CAD and CG at Zhejiang University (grant A1423) and the Joint Research Fund for Overseas Chinese, Hong Kong, and Macao Young Scientists of the National Natural Science Foundation of China (grant no. 61328204). We would like to thank the reviewers for their constructive comments and suggestions.

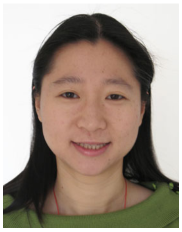
REFERENCES

1. Ben-Akiva M, Koutsopoulos HN, Yang Q. Users guide for MITSIMLab and road network editor, 2, 2002.
2. Quadstone Paramics. Quadstone paramics v4.2 analyser reference manual, 3, 2003.
3. SUMO. September 1, 2014. http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.
4. VISSIMCOM PTV. Vissim 5.30-05 user manual, 11, 2011.
5. Wilkie D, Sewall J, Lin MC. Transforming GIS data into functional road models for large-scale traffic simulation. *IEEE Transactions on Visualization and Computer Graphics* 2012; **18**(6): 890–901.
6. Sewall J, Wilkie D, Merrell P, Lin MC. Continuum traffic simulation. *Computer Graphics Forum* 2010; **29**(2): 439–448.
7. Shen J, Jin X. Detailed traffic animation for urban road networks. *Graphical Models* 2012; **74**(5): 265–282.
8. Lu X, Chen W, Xu M, Wang Z, Deng Z, Ye Y. AA-FVDM: an accident-avoidance full velocity difference model for animating realistic street-level traffic in rural scenes. *Computer Animation and Virtual Worlds* 2014; **25**: 85–97.
9. Lu X, Xu M, Chen W, Wang Z, Deng Z. A personality model for animating heterogeneous traffic behaviors. *Computer Animation and Virtual Worlds* 2014; **25**: 363–373.
10. Yang Q, Koutsopoulos HN. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies* 1996; **4**(3): 113–129.
11. Guggenheimer H. Computing frames along a trajectory. *Computer Aided Geometric Design* 1989; **6**: 77–78.
12. Siltanen P, Woodward C. Normal orientation methods for 3D offset curves, sweep surfaces and skinning. *Computer Graphics Forum* 1992; **11**: 1992.
13. Shifrin T. *A First Course in Curves and Surfaces*. University of Georgia, Athens, GA, USA, 2014.
14. Klok F. Two moving coordinate frames for sweeping along a 3D trajectory. *Computer Aided Geometric Design* 1986; **3**: 217–229.
15. Wang W, Jttler B, Zheng D, Liu Y. Computation of rotation minimizing frames. *ACM Transactions on Graphics* 2008; **27**. Article No. 2, 1–18. DOI:10.1145/1330511.1330513.
16. Xu W, Yao W, Zhao H, Zha H. A vehicle model for micro-traffic simulation in dynamic urban scenarios. In *IEEE International Conference on Robotics and Automation*, China, 2011; 2267–2274.
17. Werling M, Ziegler J, Kammel S, Thrun S. Optimal trajectory generation for dynamic street scenarios in a Frenet frame. In *IEEE International Conference on Robotics and Automation*. IEEE: Anchorage, Alaska, USA, 2010; 987–993.
18. Sharipov RA. *Course of Differential Geometry*. Bashkir State University: Ufa, Bashkortostan, Russia, 1996.
19. Wang H, Kearney J, Atkinson K. Arc-length parameterized spline curves for real-time simulation. In *5th*

International Conference on Curves and Surfaces, Saint-Malo, France, 2002; 387–396.

20. Saalfeld A. Topologically consistent line simplification with the Douglas–Peucker algorithm. *Cartography and Geographic Information Science* 1999; **26**: 7–18.
21. Hua W, Tianlu M, Xingchen K, Zhaoqi W. An all-in-one efficient lane-changing model for virtual traffic. *Computer Animation and Virtual Worlds* 2014; **25**(3-4): 385–393.

AUTHORS' BIOGRAPHIES



Tianlu Mao was born in 1977. She received her PhD degree in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2009. She is currently an associate professor of the Institute of Computing Technology, Chinese Academy of Sciences, and a member of China Computer Federation. Her current research interests are modeling and simulation.

Hua Wang was born in 1982. She received her PhD degree in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2015. She is currently an associate professor of Zhengzhou University of Light Industry, Zhengzhou, China. Her main research interests include traffic animation and environment modeling.

Zhigang Deng received the BS degree in mathematics from Xiamen University, China, the MS degree in computer science from Peking University, China, and the PhD degree in computer science from the Department of Computer Science, University of Southern California in 2006. He is currently an Associate Professor of Computer Science at the University of Houston (UH) and the founding director of the UH Computer Graphics and Interactive Media (CGIM) Lab. His research interests include computer graphics, computer animation, virtual human modeling and animation, and human computer interaction. He is the recipient of a number of awards including Google Faculty Research Award, Texas Norman Hackerman Advanced Research Award, University of Houston Teaching Excellence Award, NSFC Oversea and Hongkong/Macau Young Scholar Collaborative Research Award, and ACM ICMI Ten Year Technical Impact Award. Besides the CASA 2014 Conference General co-chair and SCA 2015 Conference General co-chair, he also serves as an Associate Editor of Computer Graphics Forum, and Computer Animation and Virtual Worlds Journal.

Zhaoqi Wang was born in 1966. He is currently a researcher, a director of PhD students of Institute of Computing Technology, Chinese Academy of Sciences, and senior member of China Computer Federation. His research interests include virtual reality and intelligent human-computer interaction.