Automatic Generation of Large-scale 3D Road Networks based on GIS Data

Hua Wang  Yue Wu  Xu Han  Mingliang Xu*  Weizhe Chen  Guoliang Chen

Abstract

How to automatically generate a realistic large-scale 3D road network is a key point for immersive and credible traffic simulations. Existing methods cannot automatically generate various kinds of intersections in 3D space based on GIS data. In this paper, we propose a method to generate complex and large-scale 3D road networks automatically with the open source GIS data, including satellite imagery, elevation data and two-dimensional(2D) road center axis data, as input. We first introduce a semantic structure of road network to obtain high-detailed and well-formed networks in a 3D scene. We then generate 2D shapes and topological data of the road network according to the semantic structure and 2D road center axis data. At last, we segment the elevation data and generate the surface of the 3D road network according to the 2D semantic data and satellite imagery data. Results show that our method does well in the generation of various types of intersections and the high-detailed features of roads. The traffic semantic structure, which must be provided in traffic simulation, can also be generated automatically according to our method.

1 Introduction

Three-dimensional(3D) virtual road network can give a high immersive visualization. It is important for traffic simulations and intelligent traffic controls [5, 11, 24, 25, 35]. The following two parts are crucial in a large-scale 3D virtual road network generation: 1) a realistic 3D scene with smooth road surface; 2) a traffic semantic structure of road network. How to automatically generate a realistic large-scale 3D road network including realistic scene, smooth road surfaces and traffic semantic structures based on the existing Geographic Information Systems (GIS) data (such as satellite imagery, elevation data, road network vector data) is a key point for immersive and credible traffic simulations [4, 28, 22].

There are mainly two methods focusing on generating realistic 3D roads. One generates scenes by combining 3D models (road data, environment data and so on) into a unified view [10, 17, 8]. In order to generate a realistic scene, the data of those 3D models can be collected by some data acquisition devices, such as airborne tilt photography equipment [12]. However, there are still lots of difficulties for data denoising and texture reconstruction. The other method combines common GIS data to generate 3D roads [9, 26]. The satellite imagery is used as texture of roads. In this way, the generated scenes are based on the real data, so they are realistic enough. Nevertheless, it is difficult to automatically generate various kinds of intersections in 3D space. In addition, both of two methods cannot generate the road structure that can be used for traffic simulation well directly.

To tackle the aforementioned challenges, in this paper, we propose an automatic method to generate large-scale 3D virtual road networks based on GIS data. We first introduce a semantic description of the large-scale road networks, and then generate the 2D shape of the road network automatically according to the semantic description and geometric design standards of roads. At last, we segment the elevation data based on the 2D shape and use piecewise B-spline curves with monotone curvature in the Frenet frame to fit the elevation curves of roads. With the open source GIS data, including...
satellite imagery, elevation data and 2D road center axis data as input, our method can automatically generate a large-scale 3D virtual road network with realistic environment and high-detailed road surfaces. Traffic semantic structures such as connectivity and adjacency relationship and so on can also be generated automatically according to our method.

To sum up, the contributions of our work are summarized as follows:

- A method of creating 3D scenes automatically with only open source GIS data (satellite imagery, elevation data and 2D road center axis data) as input. It is worth mentioning that our method supports for generating diverse and complex 3D intersections automatically.
- A way for segmenting and smoothing elevation data, so that the 3D road network has high-detailed road surface and it is also in accordance with the code of civil engineering.
- A semantic structure for 3D road network that can be used in traffic simulation.

For the rest of the paper, we first review the literatures of 3D scene modeling and traffic network modeling in Section 2, and in Section 3 we introduce our method in detail. In Section 4, we propose a traffic semantic structure for traffic simulation. At last, we discuss the method and draw some conclusions in Section 5.

2 Related work

Our work relies on two parts: One is 3D scene modeling and the other is traffic network modeling.

2.1 3D scene modeling

Using 3D models to generate 3D scenes is a traditional method. Garcııaddeo et al. [10] created roads automatically in 3D scenes by taking the attributes of lane number, intersections and the shape of roads as the input of the program. Parish et al. [17] used the software named CityEngine to create expressways and lanes in a large-scale 3D urban scene. In the above scenes, objects such as buildings, trees and roads could be displayed with detailed features, but textures were difficult to be consistent with the real environment. Dias et al. [8] used laser scanner to carry out the 3D modeling of real world scenes, which can truly and accurately restore the real world, but it is still not easy in data denoising and texture reconstruction.

The more effective method of generating 3D road network is based on the GIS data, such as satellite imagery and elevation model. 3D scenes in [20, 21, 18, 36] are created by merging vector data and elevation data. In these scenes, roads are displayed in a primitive way, without width, texture and boundary. In the given 3D scenes, GaLin et al. [9] discretized the shortest path created by them to generate piecewise clothoid, and then combined with terrain and obstacles so that the roads with slope are created. Wang et al. [26] extracted road center axes from simple attribute remote sensing images, and then combined them with terrain to create 3D road network. The roads in the above two scenes both had the width and textures, but did not support the generation and display of intersections. Wu et al. [29] defined a large-scale multi view 3D scenes, which combined 2D road panoramic map, satellite map texture and 3D city model. Li et al. [14] reconstructed a 3D scene based on image rendering. They detected the road area from the input image sequence, and then created the road network according to the road boundary control points. However, the roads in [14, 29] had no height, so that they could not show the changes of slopes. Besides, they did not generate the road network topology data for traffic simulation, either. Bruneton et al. [3] improved the elevation data through combing it with vector data to generate roads with slope fluctuation conforming to the specifications. However, there is still a gap in lanes and texture between the virtual scene and the real world. Besides, the types of intersections supported by Bruneton et al. [3] are also limited.

In short, existing methods can generate 3D realistic scenes, nevertheless, the methods are difficult to automatically generate road details, especially in mountain roads and various kinds of intersections in a large-scale scene generation.

2.2 Traffic network modeling

According to the demand of traffic simulation technology, there are many traffic network models to realize traffic simulations and path navigation. The existing traffic simulation software [34] mostly uses Node, Link, Segment and Lane to describe the traffic simulation network semantics. These
models mainly define the semantic structure of traffic network from the topological data. Wilkie et al. [27] proposed the arc representation method of Lane vector data, which realized the transformation of rough, low-detailed GIS traffic network data into high-detailed traffic network data. Mao et al. [21] proposed a method of using compressed point column to represent Lane, which not only keeps the details of traffic network, but also improves the efficiency of obtaining vehicle location information in the process of vehicle motion simulation. Based on the work of [27] and [21], Wang et al. [23] proposed a hierarchical traffic network semantic model for vehicle group animation simulation, and the route conflict relation can be generated automatically, which greatly reduces the amount of data input. In the above models, there is a close coupling relation between lanes. If the direction of a few lanes is changed, the system needs to recalculate the topological relationship between all lanes in the traffic network, which is tedious and huge. Moreover, intersection data need to be input, which is difficult to get directly from the existing GIS data.

Inspired by the above traffic models, we introduce a traffic semantic structure description and generate traffic data according to it automatically in this paper.

3 Methodology

Our method generates a realistic 3D road network automatically based on 2D road center axis data, satellite imagery and elevation data. The framework of our method is shown in Fig.1. There are mainly three processes to generate a 3D road network: semantic descriptions of the road network, semantic data generation in 2D space and 3D road surfaces generation. Firstly, we define the semantic structure of 2D road network in section 3.1 according to the road center axis data. Then we generate the 2D road shape data in section 3.2. In section 3.3, in order to obtain the elevation data of road area, we segment the elevation data with the help of 2D shape data of Link and Intersection which defined in semantic structure. After that, we smooth the elevation data of road area to get the flat and smooth 3D road surfaces.

3.1 Semantic description of the road network

In order to generate a realistic road network, we introduce a semantic structure of road network (Fig.2). There are four elements in the semantic structure.

Road is the road network in road scenes. Its topological data is 2D road center axis data. It includes Links and Intersections.

Link is used to describe the half of a road segment. It includes the topological data and geometric data. The topological data contains the connections and adjacency relations between Links. The geometric shape is its boundary. We use curves to describe it.
Figure 2: The semantic structure of roads.

**Seg_AXIS** is the center axis of a road segment. It is generated according to the road center axis. It is the junction of two neighboring Links, which describe the left and right half of the road segment. In our method, the Seg_AXIS connects the two neighboring Links. It is easy to prove that one Link has only one Seg_AXIS.

**Intersection** refers to a road junction. It is the junction of the Links that have connection relations in 3D space. It also includes the topological data and geometric data. Topological data includes the relationship between the Intersection and Links. Geometric data refers to the boundaries of the Intersections. We use a polygon area enclosed by a sequence of points to describe it.

### 3.2 Semantic data generation in 2D space

Semantic data generation in 2D space includes the generation of the 2D shapes and topological data of all the Links and Intersections in the road network. Here we show the process of how to generate them according to 2D road center axes data and the width of roads. We also generate their topological data in the process.

Considering that open source road center axes are usually represented as 2D polyline point sets, which are usually unsmooth, here we firstly use spline interpolations to generate smooth 2D vector data. Then we generate the 2D shapes of all the Links and Intersections of the road network automatically according to the following steps:

**Step1:** Translating a road center axis along its inner and outer normal directions to generate the two outer boundaries of roads. The distance of the translation is $w$. Here $w$ is the whole width of the road. Now the road network in a junction becomes several road segments (there are four segments in Fig.3(b)).

**Step2:** Generating the circular curve in the junction of two boundaries. The circular curve is tangent to those two boundaries. Its radius $R$ is defined as follows (red curves in Fig.3(c)):

$$R = \frac{v^2}{127(u + v)}$$

where $R$ is the radius of the circular curve. $v$ is the design speed of the road, $u$ and $v$ are constant. $u$ represents side-way force coefficient.

Then we collect some points of tangency to generate an Intersection. The point set $P$ in an Intersection must satisfy the following conditions:

1) The circular curve’s two points of tangency belong to one Intersection.

2) If $p \in P$, then $\exists q \in P$, and $|p - q| < L_{dis}$. Here $|\cdot|$ describes the distance between two points. $L_{dis}$ describes the size of an Intersection. In this paper, we let it be a constant (30 m).

**Step3:** For each end-point of circular curve, let it be a start-point and generate a line segment between two outer boundaries of the road along the normal of its center axis. If there are two line segments in one road segment, we delete the one which is closer to the boundaries of the Intersection and save the one that further to the boundaries (dotted lines in Fig.3(d)). Simultaneously, the start point of
Figure 3: The generation process of an Intersection and Links. (a) shows the input road center axes. In (b), we translate the road center axes to get the outer boundaries of roads according to the road width. (c) generates the circular curve. In (d), we generate the tangent lines of road center axes from the end-points of circular curve, and choose the further one as the boundaries of Intersection. (e) generates the shape of Intersection. (f) generates the shapes of Links.

the deleted line is removed from the Intersection and the end point of the saved line is added to the Intersection.

Step4: Sorting the points in the Intersection with a clockwise rotation and then connecting them to generate a polygon. It is easy to prove that the polygon is a convex polygon. We let it be the boundary of the Intersection (see Fig[3(e)].

Step5: Saving the road axes that out of polygons of Intersections and generating the Seg_Axes. We let the area of a region bounded by a Seg_Axis, the boundary of a road segment and two edges of polygon of Intersections be a Link. The boundary of road segment is also the Link’s boundary, and the Seg_Axis is also the Link’s Seg_Axis. The generated Link belongs to those Intersections. It is easy to prove that the boundary of the Link is translated, because the Seg_Axis and the curves of them are parallel.

In order to generate 3D road surfaces, we adopt the curve compression discretization algorithm (Douglas Peucker Algorithm) to discretize and compress the boundary curves of Link, and then get the following descriptions of the geometric shape data of Links and Intersections:

**Link geometric shape data:** It is used to describe the boundary data of a Link. If $A_i$ is the geometric data of the $i$th Link, it can be expressed as a closed polygon generated from the following point set,

$$\{q_{i1}, q_{i2}, \ldots, q_{iM_i}, \tilde{q}_{iN_i}, \tilde{q}_{i(N_i-1)}, \ldots, \tilde{q}_{i1}\}$$  \hfill (2)

where $q_{i1}, q_{i2}, \ldots, q_{iM_i}$ represents the point column of the $i$th Link’s Seg_Axis, $\tilde{q}_{i1}, \tilde{q}_{i2}, \ldots, \tilde{q}_{iN_i}$ denotes the boundary of the road segment, and $M_i$ and $N_i$ are their total number of points, respectively.

**Intersection geometric data:** If the current $i$th Intersection aggregates the number of road center axes $k_i$ , obviously, it can be proved that $2 \times k_i$ Links are generated through the extension of the $k_i$ roads. Then we sort these Links clockwise. We use $n_{ij}$ ($j \in [1, 2 \times k_i]$) to represent the $j$th Link, and the geometric data of the current $i$th Intersection can be defined as follows,

$$\{\tilde{q}_{n_{ij}s_{n_{ij}}}, \tilde{q}_{n_{ij}s_{n_{ij+2}}}, \ldots, \tilde{q}_{n_{ij}s_{n_{ij+2k_i}}}\}$$  \hfill (3)

where $\tilde{q}_{n_{ij}s_{n_{ij}}}$ ($j \in [1, 2 \times k_i]$) represents the coordinates of the points in the $j$th Link boundary. If the road center axis of $j$th Link end at this Intersection, then $s_{n_{ij}} = N_{n_{ij}}$, and $N_{n_{ij}}$ is defined in Eq.2; else if the road center axis start at this Intersection, then $s_{n_{ij}} = 1$.

In this paper, we use a symbolic network to define the relation between an Intersection and all of the Links that connect to it. The relation is defined as two types, positive relation and negative relation.
For two Links that owns the same Seg_Axis, if one connects to the Intersection, it is easy to prove that the other one also connects to the same Intersection. We sort them by their boundaries with a clockwise rotation. The relation between the first one and the Intersection is positive, which is represented by "+". The other is a negative relation, which is indicated by "-". This symbolic network description method is suitable for representing the intersecting relation between various types of Intersections and Links.

3.3 3D road surface generation

Here we show how to generate a 3D road surface by combining the above 2D road shape data and elevation data. The elevation data in this paper is a gray-scale image with spatial reference system. The gray value represents the elevation value at the current 2D spatial position. We first segment the elevation data of road areas based on the geometry of Links and Intersections, and then use the segmented areas to smooth road surfaces. After that, a 3D road network with high-detailed road surface is generated.

3.3.1 Elevation data segmentation

The elevation image area is represented as $\Omega$, and it is a closed area bounded by the following points,

$$\{ r_i | i \in [1, 4] \}$$  \hspace{1cm} (4)

where $r_i$ ($i \in [1, 4]$) represents the vertices of the quadrilateral image. We use $\Omega_1$ to define the region enclosed by all Links boundaries and Intersections boundaries obtained in Section 3.2, and it can be expressed as follows,

$$\Omega_1 = \bigcup_{i=1}^{\text{Link\_totalnum}} A_i \bigcup_{j=1}^{\text{Intersection\_totalnum}} B_j$$  \hspace{1cm} (5)

where Link\_totalnum is the total number of Links and Intersection\_totalnum is the total number of Intersections. $A_i$ ($i \in [1, \text{Link\_totalnum}]$) and $B_j$ ($j \in [1, \text{Intersection\_totalnum}]$) represent the boundary of the i-th Link and j-th Intersection, respectively (defined in Section 3.1). Then we split the $\Omega$ based on the $\Omega_1$, and $\Omega$ can be defined a following equation,

$$\Omega = \Omega_1 \bigcup \bar{\Omega}_1$$  \hspace{1cm} (6)

where $\bar{\Omega}_1$ represents the complement set of $\Omega_1$. Obviously, it can be proved that $\Omega_1$ represents the elevation data of road areas in the scene, and any sub area of $\Omega_1$ is the closed polygon area enclosed by the boundary of Intersection or the Link.

3.3.2 Smooth processing of road surfaces

This part introduces the approach of smoothing the road surface area $\Omega_1$. For the elevation value $z$ at any position (including the boundary) in the closed polygon area $B_i$ ($i \in [1, \text{Intersection\_totalnum}]$), it is equal to the average value of the elevation value of all the vertexes on the Intersection boundary. Then, $\forall (x, y, z) \in B_i$,

$$z = \frac{1}{2 \times k_i} \sum_{n=1}^{n=2 \times k_i} h_{in}$$  \hspace{1cm} (7)

where $h_{in} (n \in 2 \times k_i)$ is the elevation value of the n-th vertex in the i-th Intersection($k_i$ is defined in 3.2).

As for the closed polygon area $A_i$ ($i \in [1, \text{Link\_totalnum}]$), if the elevation of each point in the interior $A_i$ is assigned directly according to the download elevation data, the road surfaces would be uneven. In order to avoid this problem, we smooth the elevation of the road center axes firstly, and then smooth the elevation of other positions in the $A_i$ based on the road center axes.

Smoothness of road center axis: The first step is the curve fitting of road center axis in the 2D Cartesian coordinate space. Then the curve fitting is carried out in the Frenet frame space which
is formed along the fitting curve direction and road network elevation direction to obtain the road profile curves.

In the 2D Cartesian coordinate space, after the curve fitting of the polyline points column \(q_{i1}, q_{i2}, ..., q_{iM}\) of \(A_i\), the arc length parameterized curve \(C_i\) is obtained,

\[
C_i = C_i(s) = (x(s), y(s))
\]  

(8)

\(s\) is the arc length. \((x, y)\) is the 2D coordinate of any point in the curve \(C_i\). In that way, for any arc length \(s\) on the curve, it is easy to calculate the elevation value \(h(s)\) of this position \(h(s) = h(x(s), y(s))\).

For each point on the 2D curve \(C_i\), the coordinate space is constructed with the tangent direction and the elevation direction of the road network (the direction of the external normal of the plane where the curve is located) as the coordinate axes. The coordinate space is a 2D Frenet frame space. We generate the profile curves in the Frenet frame space. According to the road design and construction standards, the profile curves of a road are the curves with monotone and constrained curvature: the curvature is less than 0.1. Here we first generate control vectors according to the above constraint in the space and then use the piecewise B-spline curve with monotone curvature to generate the profile curves.

1) Control vector generation:

Sampling points by equal distance \(\Delta s\) along the curve \(C_i\). According to the civil engineering restrictions, the slope of road usually does not exceed 0.08%. Therefore, if the absolute value of the curvature difference between two vector points is more than 0.1, the later vector will not be collected in the control vector set. Then we can get the following control vector set:

\[
(0, h(0)), (n\Delta s, h(n\Delta s)), ..., (n_{Total}\Delta s, h(n_{Total}\Delta s))
\]

where \(n_{Total} \leq \lceil S/\Delta s \rceil\), \(S\) is the total arc length of the curve \(C_i\), and \(\lfloor \cdot \rfloor\) is the rounding symbol.

2) Profile curves generation

Here we use the piecewise B-spline curve with monotone curvature to generate the profile curves. Each piece of the cubic B-spline curve is generated by the method presented by Wang et al.[6]. We fit the above point column to get the relation curve \(\tilde{C}_i\) between \(s\) and \(h\),

\[
\tilde{C}_i = h(s)
\]

(9)

according to the Eq.9, the elevation values of each coordinate point in \(q_{i1}, q_{i2}, ..., q_{iM}\) are calculated.

Smoothness within the Link geometric area: According to Section 3.1, the Link area is a closed area derived from the extension of road axis. In the Link area, the elevation value at any point is equal to its nearest point of road axis, which means that the elevation value in the direction of the road section profile is constant.

4 Implementation and application

In this section, we introduce the implementation of our method. Besides, we also show how to generate the road structure which can be used for traffic simulation directly based on the semantic structure of 3D scenes defined in 3.1.

We first generate 2D shape data of road with the road center axis data as input according to the road semantic structure. The roads are made up of three elements, Link, Intersection and Seg_Axis. Then we segment the original elevation data by using the generated 2D road shape, and get the elevation data with only road area. Every step is mutually linked and essential. After that, smoothing the elevation data of road is our key step. Here we smooth the cross section of the road and the longitudinal section of the road respectively. The orange points in Fig.5 are collected from the output experimental results under the condition that only the road cross section is flat. At this time, the road is tortuous, not in line with the reality. The blue points are collected from the output experimental results after the smoothness of road center axis. It can be seen that the road is handled better. At last, we input the satellite imagery as texture to reconstruct the 3D scenes.

Two elements (Lane and Connector) are added in traffic semantic structure. Lane describes a trajectory of vehicles along roads. There are two types according to the data generation:
**Figure 4**: The road structure for traffic simulations

**Figure 5**: The results of our approach with the process of the smoothness of road center axes (blue points) and without (orange points).

LLane describes a trajectory of vehicles in a Link. In this paper, the geometric shape data of LLane is generated by extending Seg_Axis of the Link, and the extension distance is determined by the width of the lane. Obviously, a LLane starts from an Intersection and ends to another Intersection. Here we also use “+” and “-” to describe the relation between a LLane and an Intersecion. If the LLane starts from the Intersection, the relation is “-”, otherwise, it is “+”.

ILane represents a trajectory of vehicles in an Intersection. Its topological data defines as follows:

- It starts from the end point of a LLane that has the“+” relation with the Intersection, and it also tangents to the LLane at that end point;
- It ends to the starting point of “-” LLane that has the“-” relation with the Intersection, and it also tangents to the LLane at that starting point;
- Each point on the ILane has second-order smoothness;
- There is only one ILane between an end point of one LLane and a starting point of another LLane;

According to the definition of ILane and the actual situations, it is not difficult to generate its geometric data automatically. The generation algorithm is shown in Algorithm 1.

**Connector** is the connection between two Links. It only has topological data. If there is an ILane which connects one LLane in Link1 to another Llane in Link2, then Link1 is connected to Link2.
Algorithm 1 The algorithm for I_Lane generation

//The number of L_Lanes that have "+" relation with the Intersection
int fromN;
//The number of L_Lanes that have "-" relation with the Intersection
int toN;
//M is the number of Links that have relations with the Intersection
for (i = 1; i < M; i = i + 2) do
  k = i + 1;
  num = 1;
  while (num < = M/2) do
    if (k > M) then
      k = k - m;
    end if
    fromN = num; toN = num;
    if (fromN > N1: total number of Lane in Link1) then
      fromN = N1;
    end if
    if (ToN > N2: total number of Lane in Link2) then
      toN = N2;
    end if
    Generate one I_Lane=Bezier (Vectors of the end point of the fromNth L_Lane in the ith Link and of the start point of the toNth L_Lane in the kth Link);
    if (k! = i + 1) then
      num ++;
    end if
    k ++;
  end while
end for

Figure 6: The road structure for traffic simulation. The yellow lines represent Seg_Axis. Blue lines with purple arrowhead are the L_Lanes extended from Seg_Axis. The white lines are the I_Lanes that start from "+"L_Lanes and end at "-"L_Lanes.

5 Results

This section verifies the effectiveness and efficiency of generating a large-scale 3D road network automatically by the method proposed in this paper. Firstly, the comparison between our method and some typical methods for road networks in recent years is given. After that, the calculation efficiency is analyzed in detailed. At last, we show some typical scenes generated by our method. Results showed that our method can generate large-scale 3D scenes with high-detailed road surfaces for traffic simulation when users only have satellite imagery data, elevation data and road center axes.
Figure 7: The comparison of intersection modeling. (a) shows the Intersection generated by CityEngine and Brunton. (b) illustrates the Intersection defined by Nguyen. (c) represents the Intersection modeled in Chen and Cura. The Intersection in (d) is generated by our methods.

Table 1: The feature comparisons between some road network models and our method

| Scenes     | Dimensions of Scene | Sense of reality | Generation of Intersection | Generation of Traffic data |
|------------|---------------------|------------------|-----------------------------|---------------------------|
| Brunton    | 3D                  | Virtual simulation | Semi-automatic             | Not support               |
| Cura       | 2D                  | Realistic reconstruction | Automatic                   | Automatic |
| Galin      | 3D                  | Virtual simulation | Not support                 | Not support               |
| Wang       | 3D                  | Realistic reconstruction | Not support                 | Not support               |
| Our method | 3D                  | Realistic reconstruction | Automatic                   | Automatic |

5.1 Comparison study

Our method has significant advantages in automatic generation of 3D road networks especially intersections. Detailed feature comparisons between some typical methods and our method are shown in Table 1. Comparing some methods in showing a virtual simulation for real road networks and generating complex intersections, our method can not only generate a realistic 3D road network reconstructions based on measurement data, but also generate complex intersections and traffic data automatically. There are two kinds of scenes. The scenes in [3, 9] are created by virtual simulation, the other scenes [7, 26] are generated based on the real data completely and we classified it as realistic reconstruction. Galin and Wang supported for generating neither intersections nor traffic data. Brunton could generate simple trigeminal intersection and the four corners, but they did not mention complex intersections. As for the scene in [7], it is 2D scene, so it is limited to show the slope and flatness of roads. We chose the regions in Japan and Chongqing for the experiment, and compared the results of 3D intersections and roads in Fig. 8 and Fig. 11.

Generation of intersection: Here we compare several definitions of intersections. Intersection is modelled as a square connected to neighboring rectangles with narrow triangles in CityEngine and Brunton (see Fig. 7(a)). The intersection in Nguyen is split into non-overlapping zones: traveled way and shoulder (see Fig. 7(b)), but the intersections formed by high-curvature and high-speed trajectories are ignored. Chen generates intersections by using specification templates defined in [19], but they only have X-type and T-type intersections. Cura also generates intersections by combining the crossing angle of roads, and this method is suitable for intersections with various
Figure 8: The first row shows the 3D road surfaces generated by the method of texture mapping based projection using ArcGIS. The second row figures are generated by Wang [26]. The rest of these figures are generated by our method. The curvature of these three roads increases in turn from left to right. The bottom figures show the link and intersection we generated.

3D road network: ArcGIS allows users to combine vector maps with image based information, like aerial photographic data, and to project them onto a terrain height field surface model. The quality of 3D roads is completely depends on the quality of elevation data (see the figures of first row in Fig.8). Wang [26] also noticed this problem. They considered the height of road cross section. We tested this method in urban area and country area, and we find that it performs well in urban area. However, road longitudinal section is still not smooth enough in country area (see second row figures of Fig.8). As for the 3D scenes based on 3D models, the height of roads is constant. It’s rare to see a road with a rolling slope. Compared with these methods, our method ensures the smoothness of both cross section and longitudinal section (see the figures of third row in Fig.8).
5.2 Performance analysis

After the 3D road network is generated by this method, the semantic data of traffic network is generated automatically. The efficiency of generating semantic data of traffic network is analyzed as follows. Our experiments were implemented on the desktop PC with an Intel Xeon E5-1620 CPU and two graphics cards (NVIDIA GTX 1080 Ti).

We first generate 2D shape data of road network within 10 kilometers. Fig.9 shows the relation between the number of Link in the road network and the time used in calculation. It can be seen that the calculation time increases as the number of road center axes increases.

5.3 Validation for 3D road scenes reconstruction

In order to demonstrate the effect of our method, we use it to reconstruct a large-scale 3D rural scene, including the reconstruction of Mountain roads, ramps, various types of Intersections, etc. The result is shown in Fig.10.

Mountain road: Fig.10 shows several types of mountain road. The top three subfigures are the results using texture mapping based projection. The bottom three subfigures are the results using our method. The degree of road curvature increases from left to right. It shows that the 3D road surfaces generated by our method are in line with actual road specifications, even though the road is rugged.
Figure 11: The comparison of different kinds of 3D Intersections between original data (top) and processed data (middle). The bottom figures show the link and intersection we generated.

Figure 12: I_Lanes and L_Lanes in different kinds of Intersections. (From left to right, there are x-type intersection, y-type intersection, T-type intersection and irregularly shaped intersection.)

**Intersection**: The 3D intersections generated by our method are also coincide with the actual environment. As shown in Fig.11, our road surfaces of intersections are more flat than before, no matter what kind of shape it is.

### 5.4 Validation for traffic simulation application

Besides the Link and Intersection data, we also generated I_Lane and L_Lane for traffic simulations. Fig.12 shows the results of various intersections. The purple area is the Intersection geometry shape, the yellow lines are I_Lane and L_Lane. We can see that there is a high degree of consistency between I_Lane and vehicle trajectories, and this consistency ensures the connectivity among road segments. Based on the above scenario, we further simulate the road network in this scene. Fig.13 show some traffic simulation results based on the 3D roads.

### 6 Discussion

In our method, we use only GIS data as input to generate 3D network, and the roads in the generated scenes are high-detailed. They are flat and smooth, especially in the mountain area. In the urban area, the altitude difference is tiny (as shown in the last column of Fig.11), therefore, our approach shows more advantages in countryside. Besides, we also support for generating various shapes of
intersections, and our 3D road network can be used for traffic simulation. However, there are still some limitations of our method:

- Due to the singularity of input data, we ignore the bridges, overpasses and tunnels, and they are important in urban traffic.

- The roundabout is not modelled when we generate the 2D road shape data. We will add the definition of roundabout in later scenes.

- When we generate 3D scenes, the edges of the roads are stretched because of the altitude difference between the road area and non road area. Using higher precision elevation data as input may solve this problem.

7 Conclusion

This paper proposes a method to generate large-scale 3D roads with road details by only using 2D road center axes, satellite imagery data and elevation data. By introducing a semantic description of the large-scale road networks and using image segmentations and a curves fitting method in the Frenet frame, our method generates the 3D shape of the road network and the traffic semantic structures, including the connectivity relationship, adjacency relationship and so on automatically. In the future, we would like to optimize the boundary treatment process in our 3D surface generation. Besides, we would also generate the traffic scenes according to the demand of crow simulation [13, 15, 22, 31, 33, 30].
Acknowledgments

This work is supported and funded by the National Natural Science Foundation of China (grant no.62072415,62036010, and 61822701), the Natural Science Foundation of Henan Province (grant no.02300410496), the China Postdoctoral Science Foundation (grant no.2019M662530), the foundation for the innovation of science and technology talents in higher education of Henan Province (grant no.18HASTIT020), and the special project for COVID-19 prevention and control emergency tackling of Henan Science and Technology Department (grant no. 201100312000). We would like to thank the reviewers for their helpful feedback.

References

[1] Cityengine. Home Page. Retrieved June 18, 2020, from http://www.esri.com, 2016.

[2] A. Agrawal, R. Maringanti, and R. Joshi. Geometry-based mapping and rendering of vector data over lod phototextured 3d terrain models. 01 2006.

[3] E. Bruneton and F. Neyret. Real-time rendering and editing of vector-based terrains. Computer Graphics Forum, 2008.

[4] J. Cao, J. Zhu, Q. Zhang, K. Wang, J. Yang, and Q. Wang. Modeling urban intersection form: Measurements, patterns, and distributions. Frontiers of Architectural Research, 2020. ISSN 2095-2635.

[5] Q. Chao, H. Bi, W. Li, T. Mao, and Z. Deng. A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. Computer Graphics Forum, 39(1):287–308, 2020. doi: 10.1111/cgf.13803.

[6] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang. Interactive procedural street modeling. ACM Trans. Graph., 27(3):103: 1–10, 2008.

[7] R. Cura, J. Perret, and N. Paparoditis. Streetgen:in base city scale procedural generation of streets: Road network, road surface and street objects. CoRR, abs/1801.05741, 2018.

[8] P. Dias, M. Matos, and V. Santos. 3d reconstruction of real world scenes using a low-cost 3d range scanner. Computer Aided Civil and Infrastructure Engineering, 21(7):486–497, 2006.

[9] E. Galin, A. peytavie, N. Marchal, and E. Gurin. Procedural generation of roads. Computer Graphics Forum, 29(2):429–438, 2010.

[10] I. Garcia-Dorado, D. G. Aliaga, S. Bhalachandran, P. Schmid, and D. Niyogi. Fast weather simulation for inverse procedural design of 3d urban models. Acm Transactions on Graphics, 36(2):1–19, 2017.

[11] W. Hua, H. XiaoYu, C. LiuYang, Y. JunRu, H. Li, L. Hui, Z. FuBao, Z. RuiJie, G. ZhiMin, and M. Xu. Cognition-driven traffic simulation for unstructured road networks. Journal of Computer Science and Technology, 35(4):875, 2020. doi: 10.1007/s11390-020-9598-y.

[12] T. Ku, R. C. Veltkamp, B. Boom, D. Duque-Arias, S. Velasco-Forero, J.-E. Deschaud, F. Goulette, B. Marcotegui, S. Ortega, A. Trujillo, J. P. Suárez, J. M. Santana, C. Ramírez, K. Akadas, and S. Gangisetty. Shrec 2020: 3d point cloud semantic segmentation for street scenes. Computers & Graphics, 93:13–24, 2020. ISSN 0097-8493.

[13] C. Li, P. Lv, D. Manocha, H. Wang, Y. Li, B. Zhou, and M. Xu. Acsee: Antagonistic crowd simulation model with emotional contagion and evolutionary game theory. IEEE Transactions on Affective Computing, PP(99):1–1, 2019.

[14] Y. Li, Y. Liu, J. Zhu, S. Ma, and R. Guo. Spatiotemporal road scene reconstruction using superpixel-based markov random field. Information Sciences, 507, 2019.

[15] P. Lv, Z. Zhang, C. Li, Y. Guo, B. Zhou, and M. Xu. Crowd behavior evolution with emotional contagion in political rallies. IEEE Transactions on Computational Social Systems, 2018. doi: 10.1109/TCSS.2018.2878461.

[16] H. H. Nguyen, M. Daniel, and B. Desbenoit. Realistic urban road network modelling from gis data. In Eurographics Workshop on Urban Data Modelling and Visualisation, 2016.

[17] Y. Parish and P. Muller. Procedural modeling of cities. volume 2001, pages 301–308, 08 2001. doi: 10.1145/1185657.1185716.
[18] J. She, Y. Zhou, X. Tan, X. Li, and X. Guo. A parallelized screen-based method for rendering polylines and polygons on terrain surfaces. *Computers and Geosciences*, 99(FEB.):19–27, 2017.

[19] G. Thomas and S. Donikian. Modelling virtual cities dedicated to behavioural animation. *Computer Graphics Forum*, 2000.

[20] M. Thöny, M. Billeter, and R. Pajarola. Deferred vector map visualization. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345477. doi: 10.1145/3002151.3002157.

[21] M. Tianlu, W. Hua, D. Zhigang, and W. Zhaoqi. An efficient lane model for complex traffic simulation. *Computer Animation and Virtual Worlds*, 26, 05 2015. doi: 10.1002/cav.1642.

[22] D. Tsitos and S. Polyzos. The topology of urban road networks and its role to urban mobility. *Transportation Research Procedia*, 24:482 – 490, 2017. ISSN 2352-1465. 3rd Conference on Sustainable Urban Mobility, 3rd CSUM 2016, 26 – 27 May 2016, Volos, Greece.

[23] H. Wang, X. Kang, T. Mao, and Z. Wang. A semantic model of road networks for traffic animations. *Jisuanji Fuwu Sheji Yu Tuxingxue Xuebaofjournal of Computer Aided Design and Computer Graphics*, 26(10):1818–1826, 2014.

[24] H. Wang, T. Mao, X. Kang, and Z. Wang. An all-in-one efficient lane-changing model for virtual traffic. *Computer Animation and Virtual Worlds*, 25, 05 2014. doi: 10.1002/cav.1576.

[25] H. Wang, M. Xu, F. Zhu, Z. Deng, Y. Li, and B. Zhou. Shadow traffic: A unified model for abnormal traffic behavior simulation. *Computers & Graphics*, 70, 07 2017. doi: 10.1016/j.cag.2017.07.004.

[26] Z. Wang, H. Zhang, Q. Lu, and Y. Lian. 3d scene modeling and rendering algorithm based on road extraction of simple attribute remote sensing image. In *2018 International Conference on Computer Modeling, Simulation and Algorithm (CMSA 2018)*, 2018.

[27] Wilkie, D., Sewall, J., Lin, and C. M. Transforming gis data into functional road models for large-scale traffic simulation. *IEEE Transactions on Visualization and Computer Graphics*, 18 (6):890–901, 2012.

[28] C. Wu, M. Hu, R. Jiang, and Q. Hao. Effects of road network structure on the performance of urban traffic systems. *Physica A: Statistical Mechanics and its Applications*, 563:125361, 2021. ISSN 0378-4371.

[29] Y. Wu, C. Liu, S. Lan, and M. Yang. Real-time 3d road scene based on virtual-real fusion method. *Sensors Journal IEEE*, 15(2):750–756, 2015.

[30] M. Xu, H. Wang, S. Chu, Y. Gan, X. Jiang, Y. Li, and B. Zhou. Traffic simulation and visual verification in smog. *ACM Transactions on Intelligent Systems and Technology*, 10:1–17, 11 2018. doi: 10.1145/3200491.

[31] M. Xu, X. Xie, P. Lv, J. Niu, H. Wang, C. Li, R. Zhu, Z. Deng, and B. Zhou. Crowd behavior simulation with emotional contagion in unexpected multihazard situations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018. doi: 10.1109/TSMC.2019.2899047.

[32] M. Xu, C. Li, P. Lv, W. Chen, Z. Deng, B. Zhou, and D. Manocha. Emotion-based crowd simulation model based on physical strength consumption for emergency scenarios. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–15, 06 2020. doi: 10.1109/TITS.2020.3000607.

[33] J. Xue, H. Yin, P. Lv, M. Xu, and Y. Li. Crowd queuing simulation with an improved emotional contagion model. *Science China Information Sciences*, 2019. doi: 10.1007/s11432-018-9654-6.

[34] Q. I. Yang and H. N. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C*, 4(3):113–129, 1996.

[35] A. G. O. Yeh, T. Zhong, and Y. Yue. Hierarchical polygonization for generating and updating lane-based road network information for navigation from road markings. *Int. J. Geogr. Inf. Sci.*, 29(9):1509–1533, 2015.

[36] Y. Zhi, Y. Gao, L. Wu, L. Liu, and H. Cai. An improved algorithm for vector data rendering in virtual terrain visualization. In *Geoinformatics (GEOINFORMATICS), 2013 21st International Conference on*, pages 1–4, 06 2013. doi: 10.1109/GEOinformatics.2013.6626137.