13th COTA International Conference of Transportation Professionals (CICTP 2013)

Rendering of Dynamic Road Traffic Noise Map Based on Paramics

MA Xia-lin, CAI Ming*

School of Engineering Sun Yat-sen University, Guangdong Provincial Key Laboratory of Intelligent Transportation System, Guangdong, Guangzhou 510006, China

Abstract

Traffic flow is simplified as equal-interval line source in traditional road traffic noise prediction models. The prediction results of these models are non-dynamic and can only be used to render the static noise maps. This paper performs the dynamic simulation of road traffic noise by integrating traffic micro-simulation technology, noise emission and propagation calculation, and accomplishes the dynamic rendering of traffic noise distribution based on Paramics platform. By designing and developing Paramics plug-ins, the dynamic traffic noise map is generated by real-time rendering of the spatial and temporal noise distribution data through the traffic simulation. High validation is indicated from the study case of Zhujiang New Town in Guangzhou.

© 2013 The Authors. Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of Chinese Overseas Transportation Association (COTA).

Key Words: traffic simulation; dynamic noise map; rendering of noise distribution

1. Introduction

The road traffic noise resulting from the increase of motor vehicle has serious impact on the quality of urban residential environment. Noise map (Liu & Hu, 2010) as the crucial base of accurate environment assessment and management is currently static which is mainly designed by traffic noise model such as American FHWA Traffic Noise Model (Barry & Reagan, 1978), English CRTN Model (Department of Environment and Welsh Office UK, 1975), Japanese ASJ Model (Yamamoto, 2010) etc. However these prediction models are all based on hypothesis of stable traffic flow which traffic flow is simplified as infinite-length line sound source. In the stable traffic flow, vehicles of the same type are uniformly distributed along the central lane with the same speed. In spite of convenience in calculation, these models only output equivalent sound level of periods, which eliminates the

* Corresponding author. E-mail address: caiming@mail.sysu.edu.cn
fluctuation characteristics that traffic noise varies with time. Those outputs can’t be used to render the dynamic noise maps.

The development of computer technology, especially the traffic micro-simulation technology, makes it possible to simulate dynamic traffic noise. With traffic micro-simulation software, the dynamic traffic noise prediction model traces the parameters of each simulating motor vehicle, calculates the individual and subsequently the superposed traffic noise of each vehicle. The result from this dynamic noise prediction model is of more real meaning than static noise prediction model. Coensel (2005) , using vehicle emission model, beam tracing method and GIS to analyze the influence of traffic flow dynamics on urban soundscape successfully calculated the spatial and temporal traffic noise distribution data of the area within 1km² in Gentbrugge, Belgium.

In this article, dynamic road traffic noise distribution, rendered in Paramics, is simulated based on the combination of traffic micro-simulation technology, noise emission and propagation calculation. Dynamic road traffic noise map can be generated by designing plug-in and dynamically rendering the spatial and temporal noise distribution data while simulating.

2. Calculation of Spatial and Temporal Noise Distribution Data

2.1. Paramics: Traffic Micro-simulation Software

Paramics (Parallel Microscopic Simulation) (Quadstone, 2003) , is a microscopic road traffic simulation software package developed by Edinburgh Parallel Computing Centre, England. It’s a suite of high performance tools used to model the movement and behavior of individual vehicle on urban and highway road networks. Paramics Programmer (Quadstone, 2003b) is a framework that customize features of underlying simulation model. Access is provided through Application Programming Interface (API). The capability to access and modify the underlying simulation model through API is essential that users can add customized control strategies and make extension of the simulation ability. The Programmer contains four sets of functions: Override Functions (QPO), Extend Functions (QPX), GetValue Functions (QPG), and SetValue Functions (QPS).

2.2. Noise Data Calculation

Coensel’s dynamic traffic noise prediction model which combines traffic micro-simulation, noise emission model, beam tracing method and GIS provides a method to get the spatial and temporal noise distribution data for this research. Figure 1 presents the data calculation flow. Firstly, set up urban road network in Paramics software and provide the urban buildings information of study area beforehand. Input basic information of road network and traffic flow such as OD matrix, intersection signal timing. Based on noise emission and propagation model, the noise calculation plug-in can be designed using Paramics API. This plug-in is designed to calculate the noise prediction values of each observer point. Along the simulation process, microscopic information of each individual vehicle can be obtained in every simulation second. This kind of information, containing vehicle position, speed and acceleration, is then imported to noise emission model to calculate the noise emission intensity of individual vehicle. Finally, instantaneous sound pressure level of each noise observer point in every second can be calculated by using the noise propagation model. Repeating the process above in every simulation second and we can get the spatial and temporal noise distribution data set of observer points in the study area. Then the data set $V_{in}$ can be used for dynamic noise map rendering.
The data set $V_{tn}$ mentioned above indicates the noise value of observer $n$ at time $t$. $t=1,T$, indicates a moment in $T$-length simulation period, and is in simulation second unit. $n=1,N$, indicates the $n$-th in the $N$ total noise observer points. In the mean time, the coordinate set of these points $P_n$: $(P_{nx}, P_{ny})$ should be recorded.

3. Dynamic Noise Distribution Rendering

The noise map rendering plug-in can be conducted after obtaining the spatial and temporal distribution data set. In this rendering plug-in, API is called to load the data set and render the real-time noise distribution through repeating simulation process. Rendering of the time-varying noise data set is a process that drawing each observer point with different color according to noise value at different moment and that presents the dynamic noise distribution effect. In Paramics, each individual vehicle has the same running and position state at the same moment of every repeating simulation process if road network and simulation factor remains invariant. This ensures the correspondence between rendering result and the noise data set.

3.1. Design of Rendering Plug-In

QPX and QPS functions mainly composed the plug-in of dynamic noise distribution rendering in this article. QPX defines functions that add customized functionalities to Paramics. These functions can be triggered by a large number of events i.e. the loading, saving or refreshing of network, or the start/end of each timestep etc. QPS can set or change a data value of each simulating object. Figure 2 presents the design flow of the rendering plug-in.
Define global variables

qpx_NET_postOpen():
1. Load data set
2. Coordinate Systems Transformation
3. Color Transformation

qpx_NET_second():
simulation second indicator S auto-pluses

qpx_DRW_modelView():
re-render the current noise distribution

Simulation end

Fig. 2 Design flow of rendering plug-in
At the beginning of the plug-in, we define the global variables including noise data set \( V_{tn} \), color data set \( C_{tn} \), coordinate data set \( P_{n2} \), and simulation second indicator \( S \). \( S \) signifies current simulation seconds and indicates the time \( t \) of current rendering color data. \( C_{tn} \) corresponds with \( V_{tn} \) and indicates rendering color value of different observer point at different moment. It can be generated by transforming from corresponding noise value.

The first called API function of the plug-in is \( \text{qpx}_\text{NET}_\text{postOpen}() \) which is triggered and executes when the road network is loaded. In this function we add code to load the data sets and realize transformations between different coordinates and between noise data set and color data set. Finishing loading data sets and transformation work before starting simulation can reduce corresponding computational overhead and guarantee rendering fluency in simulation process.

For the spatial and temporal noise distribution data set are in units of simulation seconds in the time domain, variable \( S \) should auto-increment in every simulation second and be the time indicator for following rendering. \( \text{qpx}_\text{NET}_\text{second}() \) can be triggered in every simulation second so we enforce the increment operation of \( S \) in this function. Subsequently, at the end of each simulation loop we call \( \text{qpx}_\text{DRW}_\text{modelView}() \) to re-render the current noise distribution according to the time indicator \( S \).

### 3.2. Transformation between Noise Data and Rendering Color Value

In this article the noise level is indicated by gradient rendering color. In Paramics, RGB color value is stored as a 32-bit integer data and signified as \( 0xaaBBGGRR \). Figure 3 presents the design flow of the transformation algorithm between noise data value and rendering color value, which gradually distributes color values from pure red \( 0xFF0000FF \) to pure green \( 0xFF00FF00 \) and sets pure yellow \( 0xFF00FFFF \) as central value.

![Fig. 3 Design flow of color transformation algorithm](image-url)
3.3. Coordinate Transformation

Since the coordinate system Paramics uses is rectangular coordinate system, the raw data set should be transformed from geographic coordinate system to rectangular coordinate when loading the longitude and the latitude data sets (Nie & Yu, etl. 2008). What’s more, the rectangular coordinate system has opposite direction on x-axis between right-hand driving and left-hand driving. The x-axis coordinate data need to be flipped to the inverse when in use of right-hand driving, otherwise the noise distribution will shift and can’t be attached well with the simulation road network.

3.4. Rendering of Dynamic Noise Distribution

Noise observer points are drew as squares rendering with different color $C_{sn}$, where $S$ is the simulation second indicator and $n$ means the n-th noise observer point. All squares need to be re-drawn at each re-rendering, following the design flow presented as Figure 4. By calling qps_DRW_colour() we can set customized color to current rendering observer square. qps_DRW_filledRectangle() is used to draw the square which centers on the corresponding noise observer point and has the length of side that is equal to the length which divides mesh of the noise observer points. Following this design flow and calling qpx_DRW_modelView() to re-render the noise distribution as simulation is running, we can visually present the dynamic distribution of urban noise that varies with the traffic flow.

![Design flow of noise distribution rendering](image_url)

**Fig. 4 Design flow of noise distribution rendering**

4. Study Case: Zhujiang New Town

To make a validation of the dynamic noise distribution rendering method presented, Zhujiang New Town, Guangzhou, China was chosen as a study area. The chosen area has the area of $1360m \times 1640m$, containing 25 roads (including 2 main roads), 16 signal intersections, 2 round intersections and 97 buildings. The area was divided into mesh with interval of 20m and generated 5727 observer points. Using the above-mentioned rendering
method, we obtained the simulation output displayed in Figure 5. The dark gray part of this figure represents the network, black part represents the buildings, and the gradient color from red to green represents the noise distribution rendering result. It can be figured that the color is redder beside the high-traffic roads. And on the low-traffic roads, high noise level only distributes around individual vehicles.

Figure 6 presents 6 local dynamic traffic noise distribution rendering maps that are time-consecutive and have time interval of 1 second. There are two heavy vehicles A and B marked as cyan squares running toward an intersection. A is the one on horizontal road and meets the green light while B is the other one that is on vertical road and meets the red light. In the process of running toward the intersection, vehicle A keeps speed invariant and has high noise distribution around. However, vehicle B slows down because of red light and the noise distribution around it gradually reduces. Observe the empty space surrounded by four buildings in the center of each graph, it can be found that, at the beginning, when two vehicles A and B are far away from the empty space, it has lower noise distribution and is rendered green. When the two vehicles driving near to the interspaces between buildings, the color of the empty space gradually becomes yellow and has higher noise distribution. That’s not only because of the adjacency of the vehicles, but also that the noise generated by vehicles A and B can directly reach the empty space. Subsequently, when vehicles A and B reach the intersection, the noise distribution of the empty space gradually turn lower again due to the blocking of the buildings between them.
5. Conclusion

In this article we present a dynamic noise map rendering method based on micro-simulation software Paramics. Based on the spatial and temporal noise distribution data generated by using the dynamic traffic noise prediction model, accompanied with writing the plug-ins that call for the API provided by Paramics, the dynamic noise distribution rendering is realized. The plug-in designed in this article contains the functions of loading noise data set, transformation between noise data set and rendering color data set and the transformation between different coordinate systems. The study case has fluently demonstrated the dynamic noise distribution of urban area and presented its validation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (51178476).

References:


