Detailed Traffic Animation for Urban Road Networks

Jingjing Shen[a], Xiaogang Jin[a,*]

[a] *State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China*

Abstract

We present a new agent-based system for detailed traffic animation on urban arterial networks with diverse junctions like signalized crossing, merging and weaving areas. To control the motion of traffic for visualization and animation purposes, we utilize the popular follow-the-leader method to simulate various vehicle types and intelligent driving styles. We also introduce a continuous lane-changing model to imitate the vehicle's decision-making process and dynamic interactions with neighboring vehicles. By applying our approach in several typical urban traffic scenarios, we demonstrate that our system can well visualize vehicles' behaviors in a realistic manner on complex road networks and generate immersive traffic flow animations with smooth accelerating strategies and flexible lane changes.

Keywords: traffic animation, microscopic model, lane change

1. Introduction

An efficient traffic transportation system is always indispensable to the functioning and prosperity of a modern, industrialized society. Due to the fast growing volume of vehicular traffic and the complex road topologies in modern cities, there are many traffic problems, such as traffic jams, incident management, signal control optimization and network design, which cannot be solved by traditional tools based on analytical methods. Many research activities have focused on modeling, simulation and visualization of modern traffic by taking advantage of the advanced computer technology, either to evaluate alternatives in traffic management or to assist traffic system construction in urban development. Additionally, along with the 3D urban visualizing tools like Google Maps and Virtual Earth, and the increasing need of constructing vivid 3D worlds in entertainment industry (mostly in 3D films and games), there is a rapidly growing demand to incorporate immersive traffic scenarios into these 3D environments.

There are massive mathematical descriptions on modeling and simulating the traffic flow, which typically can be classified into microscopic (agent-based) or macroscopic (continuum-based) models. While microscopic methods try to model the dynamics of each vehicle under the influence of its surrounding vehicles, macroscopic models treat the collection of vehicles as a continuous flow, using formulas inspired by gas-kinetic flow or hydrodynamic flow equations. Since the continuum models are mainly designed to simulate large-scale road networks in parallel, focusing on aggregate behaviors and collective quantities like spatial density and traffic flux, they are not very suitable for detailed, non-uniform traffic simulation on road system with many junctions and signalized intersections. Instead, many microscopic models are adopted for specific urban traffic simulations due to their flexibility to integrate heterogeneous agents, to generalize to diverse road topologies and to concentrate on various interactions between adjacent vehicles according to local road situations.

Two main challenges exist when simulating realistic urban traffic scenarios: automatic motion control in connection regions and natural lane-changing behaviors on multi-lane roads (particularly when approaching the critical junctions such as the merging, weaving areas and signalized crossings). The existing traffic simulators are typically designed to observe overall traffic phenomena on simple hypothetical road networks for experimenting and analyzing purposes. Therefore, the driving behaviors are highly abstract and simplified. Their control and display modules are comparatively coarsely implemented, which cannot be directly applied to produce a fine realistic traffic animation. In addition, the existing lane-changing techniques, which were proposed mainly for single-lane freeway traffic and idealized merging regions, handle the vehicle's lane change as an instant jump in one simulation step. There are few detailed 3D implementations that account for the natural continuous interactions between the involved vehicles in the dynamic lane-changing process.

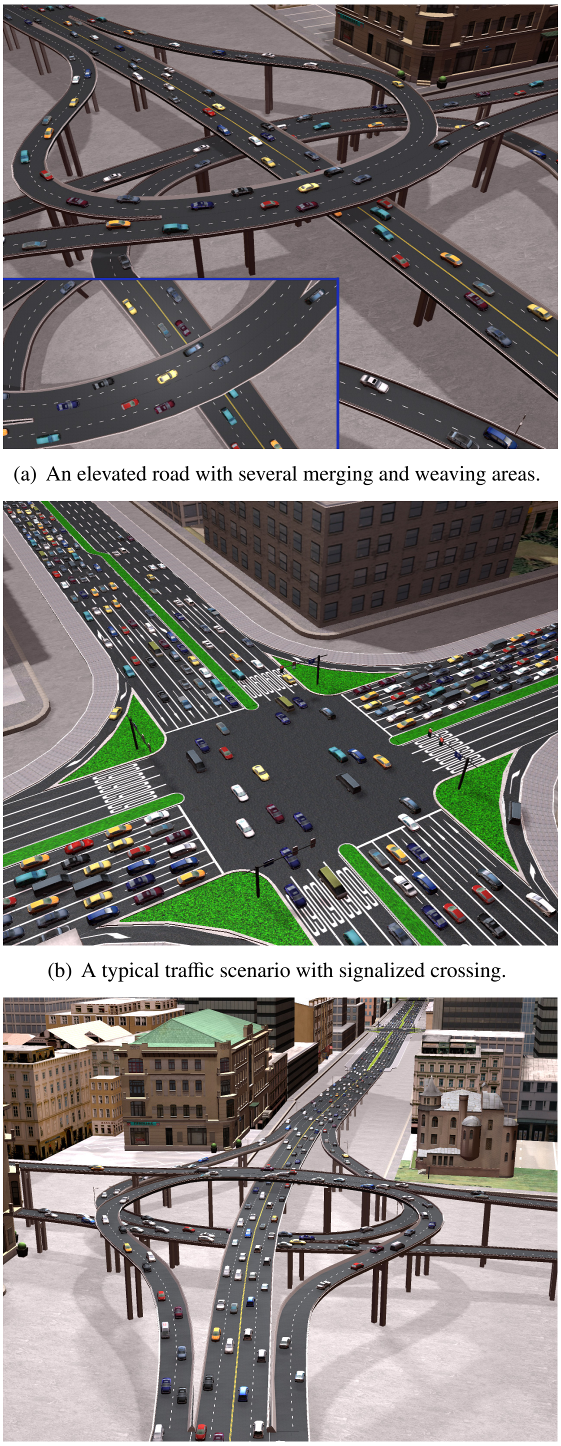
In order to produce vivid traffic animations, we present a novel microscopic traffic simulation system which incorporates several intelligent agent-based techniques. The system can visualize various traffic behaviors realistically on complex road networks with typical signalized crossing, merging and weaving areas. Two sub-models are utilized: the intelligent driver model and the continuous lane-changing model.

Our detailed implementation extends the car-following model to better handle the vehicle's reaction to other vehicle's cut-in and the stop-and-go signals in junctions, showing smooth speedup and slowing down actions in a realistic manner. We

[*] Corresponding author

Email addresses: shenjingjing@zjucadcg.cn (Jingjing Shen), jin@cad.zju.edu.cn (Xiaogang Jin) also use a flexible continuous lane-changing model to safely simulate the driver's free or imperative lane-changing behaviors in order to overtake or to get into the intended lane for turning and merging when approaching the junctions.

We have built a typical urban arterial network consisted of multi-lane roads, signalized crossings, merging junctions and elevated highways with several merging and weaving areas (see Figure 1). By applying our approach to this network, we produce an immersive traffic flow animation with realistic driving behaviors and flexible lane changes, demonstrating our system's ability to visualize complex urban traffic with a rich variety of vehicle types, driving styles and road topologies.



(a) An elevated road with several merging and weaving areas.

(b) A typical traffic scenario with signalized crossing.

(c) A global view of the road network.

Figure 1: Various traffic scenarios with different junction types.

2. Related Work

The studies and research on traffic dynamics have started since 1935. During the following decades, lots of traffic models have been proposed and extended to explain traffic phenomena, such as traffic jams and stop-and-go effects. Recently, many researchers and engineers have started to employ advanced computer technology to simulate and visualize their traffic models to obtain direct, visible results for practical applications, such as city road planning, traffic regulation and congestion analysis.

Helbing [1] reviewed the history of traffic modeling and broadly classified the modeling approaches into microscopic descriptions and macroscopic descriptions.

Macroscopic models view traffic flow as continuum dynamics like fluid or gas. The popular LWR model [2] is the first hydrodynamic model which uses a first-order partial differential equation. Newell [3], Daganzo [4] and Lebacque [5] subsequently developed several variants of the LWR model to make it suitable for more general situations and to avoid shock fronts. Payne [6] and Whitham [7] proposed a gas-like dynamic representation, a second-order system of equation known as PW model in 1970s. This model was further extended by Aw [8] and Zhang [9] to eliminate nonphysical behavior, referred as ARZ model. In computer graphics, Sewall [10] proposed a real-time large scale continuum traffic model for 3D scenes, including a discrete multi-lane changing method. Their approach describes each single lane as a separate one-dimensional flow, and the vehicle's speed is determined by the fluid cell containing it. Since the flowing speed varies little, this technique is limited to idealized connected roads and cannot cover many details such as comparatively sharp slowing down, stopping, or imperatively cutting-into adjacent lanes.

While the macroscopic models deal with collections of vehicles and are therefore computationally more efficient than microscopic ones, they are mainly used for macroscopic analytical investigations, rather than finer level of detailed simulation like the microscopic ones.

Microscopic models, also known as agent-based approaches, assume that the acceleration of a vehicle is determined by its neighboring vehicles, especially the one ahead. Pipes [11] introduced a car-following model, which was later validated by Chandler [12]. Gerlough [13] summarized a set of car-following rules in his dissertation about traffic simulation in 1955. These approaches were followed by an avalanche of extensions, such as the non-linear car-following model [14], the optimal velocity model [15], the intelligent driver model [16] [17] and the cellular automaton model [18]. These agent-based models have been tested and applied to many traffic simulation systems, such as CORSIM [19], MITSIM [20], PARAMICS [21], VATSIM [22] and SUMO [23]. In addition, Sewall [24] proposed a traffic reconstruction and visualization method from discrete spatio-temporal data detected by sensors. This method scatters the whole state-time space and searches for an approximate trajectory for each vehicle. The time and memory required by this reconstructing process grow quickly when the volume of vehicles and the discrete states increases.

The lane-changing models are introduced in traffic simulation to investigate their effects on traffic flow and also to make the simulation more realistic. Since the lane-changing behaviors often act as initial perturbations, it is crucial to simulate their impact on the capacity, stability and breakdown of traffic flows. Kesting [25] introduced the freeway based MOBIL lane-changing model which is suitable for the intelligent driver model [16], taking acceleration as a primary consideration. Hidas [26] investigated and modeled the lane-changing behaviors around the merging and weaving regions. Since these models were designed for ideal analytical purposes, they implemented the lane change action as an instantaneous jump from current lane to the target lane in one simulation update. However, to achieve realistic animation results in 3D urban scenarios, it is vital to simulate the continuous lane-changing process with a smooth curve trajectory and dynamic interactions between the involved vehicles to keep safe.

3. Overview

Our system aims to visualize various urban traffic scenarios in a realistic manner, in which the transport network is mainly composed of multi-lane roads connected by diverse junctions. The system can be divided into three components: motion instructed by follow-the-leader model, lane-changing behaviors determined by the adaptive continuous lane-changing model and transition guided by connection information in the junction regions. The vehicle's kinetic behavior is manipulated by the first two components. The last one makes the vehicles flow naturally on the whole road network.

Our follow-the-leader model utilizes Martin Treiber's freeway-based intelligent driver model [16] (IDM) and the enhanced model [17], which closely imitates a driver's response to gaps and the relative speed with his leading vehicle (the vehicle ahead in the same lane). The vehicles' accelerating and braking actions happen more frequently around the signalized intersections, merging and weaving sections. The original IDM model would lead to a slow accelerating process for the vehicles that once stopped before the signalized crossing. We make a modification on the IDM model to generate more fluent speedup around the stop-and-go regions in animation.

Inspired by the investigation of Hidas [26], we classify drivers' lane-changing behaviors into two types: free lane-changing and imperative lane-changing based on local traffic situation. The drivers' motivations include speedup and getting into a lane that allows their intended turning movement at signalized crossing or merging movement around merging (weaving) areas. The current methods either implemented the lane change as an instant jump or a predefined curve on a road map, and didn't cover all the lane-changing situations in urban road networks. Since we attempt to generate detailed 3D urban traffic animations, we propose a flexible model to create realistic lane-changing actions in various traffic situations. Moreover, we design a reliable checking procedure to estimate the safety of the continuous lane-changing process.

In our simulator, the traffic network is divided based on three basic types: the multi-lane road, unsignalized junction and signalized junction. Each road structure contains a list recording the vehicles driving on it and some basic information about lanes, speed limit and density. Both junction structures contain the descriptions about detailed lane connections between the incoming and outgoing roads. The signalized junction is implemented by simulating the functioning of traffic lights in real-life situation. When a vehicle approaches the end of the road, it checks the information in the connected junction to decide where to go and when to go.

4. Motion Control on Vehicles

In this section we discuss about the intelligent agent-based techniques we used to control the vehicle's motion. Two important sub-models are implemented in the simulator: the modified follow-the-leader model and the continuous adaptive lane-changing model. In our implementation, the existing models are more or less modified to enhance their suitability for our system to produce fast and appealing traffic animations on both freeways and signalized urban arterial networks.

4.1. Extended Follow-the-leader Model

The follow-the-leader model, the most popular used agent-based model, assumes that the acceleration of a vehicle is dependent on the front vehicle (named as its leader). The locomotion of a vehicle is usually formulated as a function of its desired velocity, the relative speed and the gap to its leader. Generally, it accelerates when their gap increases and brakes when the gap decreases. Treiber's IDM model [16] well describes each vehicle's action according to its leader's motion state, and shows a stable crash-free dynamics with an intelligent braking strategy. However, it is proposed based on freeway traffic for one-lane road simulation and would lead to a visually slow accelerating process for the vehicles that once stopped before the signalized crossing. We modified the IDM model to make it more suitable for signalized urban road networks. Figure 2 is an illustration of the involved vehicles.

In our simulator, each intelligent vehicle is equipped with the constant parameters $(v_d, s_{\min}, a_{\max}, b_{\text{com}}, T, \delta, pf)$ that describe the basic driving capability, and the dynamic motion parameters $(p(t), v(t), a(t))$ that stand for the position on the road, the velocity and acceleration of the vehicle respectively at the moment t. The constant parameters of the vehicle are initialized

Table 1: Basic vehicle parameters.

Parameter	Meaning	Common value
v_d	Desired speed	$100 \pm 20 km/h$
$s_{\min}$	Minimum safe distance	$2.5 \pm 1.0 m$
$a_{\max}$	Maximum acceleration	$3.0 \pm 1.0 m/s^2$
b_{com}	Comfortable deceleration	$2.2 \pm 1.0 m/s^2$
T	Driver's response time	$1.0 \pm 0.5 s$
δ	Free acceleration exponent	4
pf	Driver's politeness factor	$0 \sim 1.0$

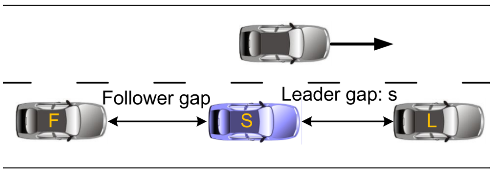


Figure 2: The relationship between the involved vehicles: the Subject (S), the Leader (L), the Follower (F) and their gaps. The leader gap s is a key factor in the subject vehicle's locomotion determination.

when it is created. Their meanings and the commonly used data ranges are shown in Table 1. Their effects on simulation result are illustrated in Table 2. The dynamic motion parameters are updated in each time step. We view each vehicle's motion as a single path in the middle of the lane, and define its lane changing trajectory as a dynamic curve between the adjacency paths.

By specifying the vehicles' individual parameters, we are capable of simulating a vast variety of vehicle types and driving styles. Each individual reacts differently based on both the hardware capacity and subjective intention, which could greatly augment the vehicles' diversity in locomotion.

The IDM acceleration is a continuous function of the current speed v, desired speed v_d, the gap s and relative speed $\Delta v = v - v_l$ to the leader. Compared with the original formulation in [16], we modified the deceleration term by adding an activation governing control part, in order to produce visually smoother reactions for the vehicles that once stopped. The modified expression is given by:

$$a_{\text{idm}} = \overbrace{a_{\max}\left[1 - \left(\frac{v}{v_d}\right)^{\delta}\right]}^{\text{free acceleration term}} \overbrace{- b_{\text{com}}\left(\frac{s^*}{s}\right)^2 (\Theta(\Delta v) + \Theta(s_b - s))}^{\text{deceleration term to keep safe}} \tag{1}$$

$$s^* = s_{\min} + vT + \frac{v\Delta v}{2\sqrt{a_{\max} b_{\text{com}}}} \tag{2}$$

The acceleration can be separated into a free road term (describing the driver's behavior to reach its desired velocity v_d) and a deceleration term (describing the driver's behavior to keep safe clearance) with an activation governing control $(\Theta(\Delta v) + \Theta(s_b - s))$:

$$a_{\text{free}} = a_{\max}\left[1 - \left(\frac{v}{v_d}\right)^{\delta}\right] \tag{3}$$

$$\begin{aligned} a_{\text{brake}}(t) &= -b_{\text{com}}\left(\frac{s^*}{s}\right)^2 (\Theta(\Delta v) + \Theta(s_b - s)) \\ &= -b_{\text{com}}\left(\frac{s_{\min}+vT}{s} + \frac{v\Delta v}{2s\sqrt{a_{\max} b_{\text{com}}}}\right)^2 (\Theta(\Delta v) + \Theta(s_b - s)) \end{aligned} \tag{4}$$

The free acceleration term is determined by the desired speed v_d, the maximum acceleration $a_{\max}$ and the acceleration exponent δ. The deceleration term is well-defined to show an intelligent braking strategy with smooth transitions between acceleration and deceleration behaviors. It is relevant to the actual gap s and the desired gap s^*. s^* is calculated by equation (2) and it consists of three terms: the minimum required distance $s_{\min}$, the braking response distance vT and the dynamic control part $\frac{v\Delta v}{2\sqrt{a_{\max} b_{\text{com}}}}$.

The natural and intelligent control exhibited by this model can be observed when considering the following situations 1)-4).

1) On a nearly empty road, where the gap to the leading vehicle is very large ($s \gg vT, s \gg v\Delta v$), the IDM acceleration is dominated by $a_{\text{free}} = a_{\max}\left[1 - \left(\frac{v}{v_d}\right)^{\delta}\right]$. The driver attempts to reach his desired speed v_d. The acceleration gradually vanishes as v approaches v_d. The change of the acceleration is controlled by the acceleration exponent δ. $\delta = 4$ has been verified as a reasonable value to produce realistic behavior in Treiber [27].

2) In traffic with small net distances, for negligible velocity difference $\Delta v \approx 0$, $v < v_d$ and small constant distance $s \approx s_{\min} + vT < s_b$, the deceleration term is approximately equal to $-b_{\text{com}}\left(\frac{s_{\min}+vT}{s}\right)^2$, which resembles a simple repulsive force such that the gap is quickly enlarged towards an equilibrium distance (characterizing by a minimum distance plus a velocity-dependent headway).

3) In a situation where the vehicle is approaching a standing vehicle or obstacle with a high approaching rate (Δv is very large), the deceleration term is dominated by $-b_{\text{com}}\left(\frac{v\Delta v}{2s\sqrt{a_{\max} b_{\text{com}}}}\right)^2 = -\frac{1}{a_{\max}}\left(\frac{\Delta v^2}{2s}\right)^2 = -\frac{a^2_{\text{shall}}}{a_{\max}}$. In such an emergency situation, drivers brake heavily to get their vehicles under control. This term will dynamically adjust the braking value used in simulation inside the maximum braking range (we use the same value with $a_{\max}$ based on the assumption that the vehicle with larger maximum acceleration capacity also has a larger maximum braking ability).

4) In a signalized crossing region, a vehicle stops and waits before the line when the corresponding traffic light is red. When the light turns green, standing vehicles need to react quickly to pass through the crossing. The original model makes vehicles accelerate very slowly with too much cautiousness. We add a new activation governing control part $(\Theta(\Delta v) + \Theta(s_b - s))$ so that the deceleration term only plays its role when necessary.

In our simulation, the deceleration part starts to play its role only when the vehicle is approaching its leader ($\Delta v \geq 0$) or the gap s is smaller than a threshold s_b ($s_b = hs^*, h \geq 1.0$ in our experiment). We use a step function $\Theta(t)$ to govern the activation. By modifying the deceleration term to react in a more relaxed way, the vehicle willaccelerate faster in the restarting process (see the velocity comparison by using the original IDM model (in green) and our modified model (in blue) in Figure 15). In the final animation result, we find that this accelerating control is close to realistic action and the vehicle's reaction is smooth when the traffic light turns green.

The dynamic control part in the deceleration term reflects the driver's smooth reaction to keep crash-free in advance. When the vehicle's speed or approaching rate is large ($v\Delta v$ is large), or the braking ability ($a_{\max}$, b_{com}) is small, this part will lead to a large desired gap, and vice versa. In emergency situations (see situation 3), it will dynamically adjust the braking response within the maximum braking range.

In most situations, updating vehicles using the above modified IDM accelerating term shows a satisfactory result with smooth accelerating and braking motions. However, when we add the lane-changing actions on multi-lane roads, the IDM acceleration would lead to the lane-changing subject's unnecessary slow down and its new follower's unrealistic sharp braking in the target lane. These overreacting behaviors are caused by the sudden drop of the actual involved gaps at the moment the subject inserts into the target lane.

To eliminate the vehicle's overreaction due to the cut-in action, we apply the method proposed by Kesting [17], which introduces a constant-acceleration-heuristic model (CAH) to deal with these unnecessary strong braking reactions. The CAH model takes the leader's current acceleration value into account and leads to zero deceleration for some cases that clearly require a moderate braking reaction. For example, in a stationary car-following situation ($\Delta v = 0, a_l = 0$), a_{cah} might be zero for arbitrary values of the gaps s and v.

The CAH model is proposed based on the following assumptions: firstly, the accelerations of the subject and the leading vehicle will not change in the relevant future. Secondly, no safe headway or minimum distance is required at any moment and drivers will react without delay (driver's response time is zero).

For the given actual values of the gap s, velocity v, velocity v_l and acceleration a_l of the leading vehicle, the CAH acceleration expression is given by:

$$a_{\text{cah}} = \begin{cases} \frac{v^2\tilde{a}_l}{v_l^2 - 2s\tilde{a}_l}, & v_l\Delta v \leq -2s\tilde{a}_l \\ \tilde{a}_l - \frac{\Delta v^2 \Theta(\Delta v)}{2s}, & \text{otherwise,} \end{cases} \tag{5}$$

where $\tilde{a}_l = \min(a_l, a_{\max})$ is used to make sure that the effective acceleration is within the subject's acceleration capability $a_{\max}$.

When calculating the maximum possible acceleration while keeping crash-free, the CAH model first predicts whether the velocity of the leading vehicle is zero or nonzero at the time when the minimum gap ($s_{\min} = 0$) is reached. When $v_l\Delta v \leq -2s\tilde{a}_l$, vehicles have stopped before $s_{\min} = 0$ is reached. The subject vehicle will try to use maximum safe acceleration and expect zero extra gap. Therefore, $\left(-\frac{v^2}{2a_{\text{cah}}}\right) - \left(-\frac{v_l^2}{2\tilde{a}_l}\right) = s$, from which we can obtain $a_{\text{cah}} = \frac{v^2\tilde{a}_l}{v_l^2 - 2s\tilde{a}_l}$. Since the negative approaching ($\Delta v < 0$) does not make sense to the CAH model, we eliminate it using $\Theta(\Delta v)$.

Our final model is based on the assumption that drivers always try to utilize the maximum crash-free acceleration. For the situations (like the stationary car-following situation) where the IDM model leads to unnecessarily strong braking reactions, the CAH acceleration is much larger and results in a more relaxed reaction. However, since the CAH model does not consider safe headway, minimum distance and response time, it cannot be directly used as a complete model (it fails on other situations). In the final model, we use the CAH model only as an indicator to determine whether the IDM model will lead to unrealistically high decelerations or not. Since the final acceleration should be a continuous and differentiable function of the IDM and CAH accelerations, the most simple functional form is given by equations (6)-(7).

$$a_{\text{adjust}} = a_{\text{cah}} + b_{\text{com}} \tanh\left(\frac{a_{\text{idm}} - a_{\text{cah}}}{b_{\text{com}}}\right) \tag{6}$$

$$a_{\text{final}} = \begin{cases} a_{\text{idm}}, & a_{\text{idm}} \geq a_{\text{cah}} \\ (1-c)a_{\text{idm}} + c a_{\text{adjust}}, & \text{otherwise} \end{cases} \tag{7}$$

Kesting [17] has experimented and verified that the weight $c \in [0.95, 1.0]$ can produce realistic actions. $c = 0.99$ is the most commonly chosen value.

In each update step, we use the final acceleration calculated from equation (7) to update the vehicle's speed and position as shown below:

$$v(t + \Delta t) = v(t) + a_{\text{final}}\Delta t \tag{8}$$

$$p(t + \Delta t) = p(t) + v(t + \Delta t)\Delta t \tag{9}$$

In order to generate real-time animation, we set $\Delta t = 0.04s$ (25 frames/second). We use the semi-implicit Euler integration to update the vehicle's velocity and position. All the simulations and experiments have showed stable results since the vehicle's acceleration value is independent of its position value.

4.2. *Continuous Lane-changing Model*

Based on Hidas's study [26] and our visual simulation purpose, we roughly divide the lane-changing behaviors on urban roads into two categories: free lane-changing and imperative lane-changing. The first type frequently happens in a comparatively free road condition, in which the gaps between the involved vehicles are large enough for safe lane change, indicating that the subject vehicle's cut-in action happens independently and there is little interference between the subject and follower vehicle. This kind of lane change is normally triggered by the subject's intention to take the speed advantage in the target lane. The other type is more complicated since we need to model the involved vehicles' interactions due to the local congested situation. It is applied when the subject vehicle demands a lane-changing action because of some imperative factors such as end of lane or intended turning movement around the crossing (Figure 3), while the current gaps are not large enough for free lane change.

Unlike 2D simulators that mainly aim at observing traffic flows and abstract the lane-changing process as an instant jump in one single simulation update, the lane-changing behaviors in

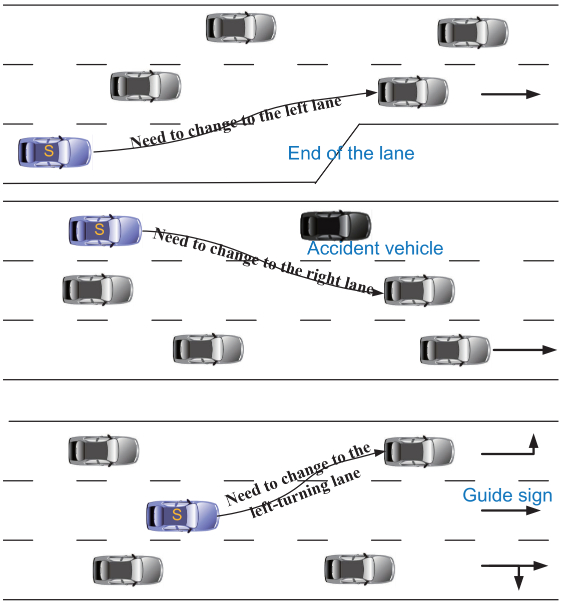

Need to change to the left lane
S
End of the lane
S
Need to change to the right lane
Accident vehicle
Need to change to the left-turning lane
S
Guide sign

Figure 3: Typical mandatory situations where a vehicle must change its lane.

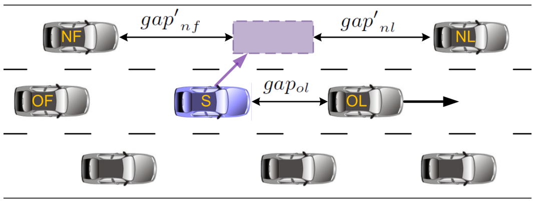
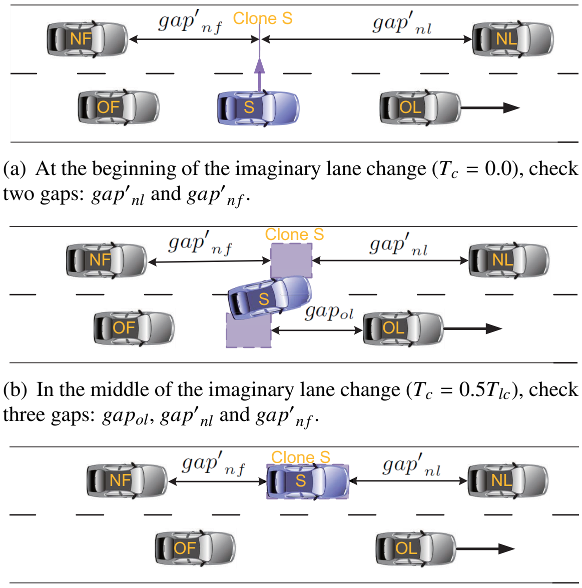


Figure 4: An illustration of free lane-changing process.



(a) At the beginning of the imaginary lane change ($T_c = 0.0$), check two gaps: gap'_{nl} and gap'_{nf}.

(b) In the middle of the imaginary lane change ($T_c = 0.5T_{lc}$), check three gaps: gap_{ol}, gap'_{nl} and gap'_{nf}.

(c) At the end of the imaginary lane change ($T_c = T_{lc}$), check two gaps: gap'_{nl} and gap'_{nf}.

Figure 5: In the estimating procedure, three critical moments are estimated to decide whether a possible lane change would be safe based on an imaginary lane-changing process.

our system is a continuous process with a safety pre-checking procedure. The final visual lane-changing trajectory is a curve determined by the involved vehicles' speeds and gaps. Compared with those 3D methods proposed by Sewall et al. [10], [24], our main benefit is that we employ the imperative lane change to deal with typical mandatory situations where a vehicle must change its lane (Figure 3). As shown by our supplementary videos, many cooperative lane-changing actions take place naturally in these situations.

Kesting's MOBIL model [25] is a general free lane-changing model proposed for a wide class of car-following models. It evaluates the utility of a given lane and the risk associated with lane changes in terms of accelerations calculated with microscopic models. We incorporate our checking procedure to this model and implement continuous lane changes. To visualize the imperative lane-changing behaviors, we simplify Hidas's interaction model [26] and implement the cooperative process by two steps: the new follower's deceleration to give way and the subject's adjustment to cut into a suitable gap.

4.2.1. Free Lane-changing Action

We extend Kesting's double-lane MOBIL model [25] to implement the multi-lane free lane-changing actions. The estimation process includes the safety check and the incentive comparison. We first compare the acceleration the subject vehicle could get in current lane and its adjacent lanes in the next time step, and weigh the possible advantages (self speedup) and disadvantages (others slow down) of the lane-changing behavior. If the benefit is larger than a preferred threshold and the gaps are safe enough, the subject vehicle will move from its current lane to the more beneficial target lane (Figure 7(a)).

Figure 4 shows an illustration of free lane-changing process. The checking procedure on involved gaps during an assumed imaginary lane-changing process is demonstrated in Figure 5.

Take the left lane-changing process as an illustration (the right lane-changing is similar). The motivation and requirement of free left lane-changing process are given by:

$$(a) \quad a'_{nf} \geq -b_{nf}$$

$$(b) \quad gap'_{ol} \geq gap'_{\min}(s, ol, T_c)$$

$$gap'_{nl} \geq gap'_{\min}(s, nl, T_c) \qquad gap'_{nf} \geq gap'_{\min}(nf, s, T_c)$$

$$gap_{\min}(a, b, T_c) = s_{\min}(a) + w_1\,(s^*(a) - (v(a) - v(b))\,T_c)$$

$$(c) \quad \Delta adv_{\text{left}} = \underbrace{a_s{}' - a_s}_{\text{subject}} + pf\left(\underbrace{a'_{nf} - a_{nf}}_{\text{new-follower}} + \underbrace{a'_{of} - a_{of}}_{\text{old-follower}}\right) \geq \Delta a_{thr}$$

All the accelerations are calculated using our follow-the-leader model, and the variables with superscript ($'$) are calculated based on the assumption that the lane-changing action had already happened.

Condition (a) is the safety criterion and guarantees that after the lane change, the deceleration of the new follower a'_{nf} in the target lane does not exceed its preferred comfortable value.

Condition (b) estimates the safety of the continuous lane-changing process by checking the gaps between the affected vehicles at three imaginary moments: $T_c = 0.0, 0.5T_{lc}, T_{lc}$, where T_{lc} stands for the total time needed for the subject to complete the lane change (in inverse proportion to the subject's speed). See Figure 5. Theoretically speaking, the more moments we check, the more accurate the estimation would be. However, since all these checks are based on inaccurate predictions by using constant accelerations, it is not worth spending much time on checking numerous moments. By simulation, we find that in most cases the lane-changing behavior would be safe when the gaps at these three moments are large enough. However, collision may happen in regions that close to the merging point and signalized crossing when the velocities of the involved vehicles are quite uneven. We solved this problem by checking several additional moments during the process.

The minimum clearance gap to vehicle b required by vehicle a at moment T_c during an imaginary lane change is calculated by $gap_{\min}(a, b, T_c)$. We need to consider the vehicle's desired gap and politeness to show the intelligent and diversity. Since the desired gap is only calculated (using equation (2)) and accurate at the beginning of the imaginary lane change, we should further predict the gap variance at moment T_c (expressed as $(v(a) - v(b))\,T_c$, which we assume that the vehicle should tolerate). Therefore, we design this equation with three parts: the driver a's minimum safe distance $s_{\min}$, the predicted desired gap with variance tolerated at moment T_c: $(s^*(a) - (v(a) - v(b))\,T_c)$ and his extra space preference $s_p = pf * s_{\min}$. w_1, w_2 are weighting factors.

To calculate the future gaps in an imaginary lane change, we make the following assumptions. Firstly at the evaluating moment $T_c = 0.0$, we insert a clone of the subject vehicle into the target lane assuming the clone vehicle's length is zero (Figure 5(a)). The clone vehicle's parameters are identical with the subject except the vehicle length. The subject's acceleration value is then updated according to its clone's relationship to the new leader using equation (7). Secondly at moment $T_c = 0.5T_{lc}$, we change the subject vehicle's length in original lane and its clone vehicle's length in the target lane to half of the vehicle length (Len_s) (Figure 5(b)). Finally, we change the clone vehicle's length to Len_s and check the safety after the lane change(Figure 5(c)).

Condition (c) is the incentive criterion that estimates whether a lane change will improve the subject's local situation, typically measured in acceleration values. The first term denotes the utility of a possible lane change for the subject, and the other terms indicate the influence on the involved neighbors (the original and new followers). The politeness factor pf (normally $\in [0, 1]$) determines the degree the subject driver concerns about the followers that might be influenced when making a lane-changing decision. We set Δa_{thr} to make a certain inertia and prevent lane changes for marginal advantage.

The considered lane change is favorable if the three conditions are all satisfied. If both left and right lane changes are favorable, we choose the one with larger advantage as the final lane change direction. Note that, the value of pf can be interpreted as the degree of altruism. It is interesting to notice that when pf is large enough, the subject may change its lane even if this change will lower its speed, just in order to benefit its original follower. This giving way situation can be observed in realistic scenarios when the follower urges the slower subject to speed up too often.

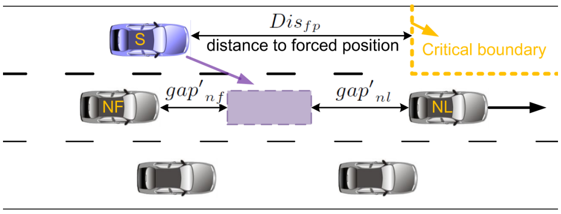


Figure 6: An illustration of imperative lane-changing process.

In conclusion, the free lane change aims at increasing overall accelerations of involved vehicles, not only the subject. In daily road traffic, vehicles are always likely to stay in their lanes for a while after a lane-changing behavior. Hence, we add a waiting interval for each vehicle, forcing it to stay in its lane for some time. Furthermore, we notice that city drivers always prefer choosing a sparser lane for long term consideration. To simulate this tendency, we could put the lane's traffic density as an additional consideration when evaluating the advantage of a possible lane-changing.

4.2.2. *Imperative Lane-changing Action*

We utilize the simplified imperative lane-changing model to imitate the interactions between the involved vehicles when the subject needs to change its lane but the interspaces around the local road do not allow for free lane change. Since our system aims at visual simulation of massive urban traffic, we pay more attention to the visual effects rather than model accuracy (compared with Hidas's model [26]). We simplify this cooperative process as the new follower's slowing down to make larger gap for cut-in and the subject's reaction to catch up the new leader (Figure 7(b)). Figure 6 shows an illustration of imperative lane change.

The imperative lane-changing is typically motivated by mandatory lane-changing situations (Figure 3), large speed advantage or queue advantage (related to lane density). Therefore, the forced position can be the end of the lane, the merging point, the waiting line before the signalized crossing or a position close to the leader vehicle where the subject insists to change its lane for large speed or queue advantage.

The feasibility is measured by the new follower's willingness to slow down. The safety checking procedure is similar with the free lane change (Figure 5), except that the minimum acceptable gaps ($gap_{\min}(a, b, T_c)$) are much smaller than in free lane change case and the new follower would decelerate for a period. In fact, it is safe to set all required $gap_{\min}(a, b, T_c)$ to $s_{\min}$. However, in order to augment the diversity of imperative lane change behaviors, we still use the equation of $gap_{\min}(a, b, T_c)$ described in section 4.2.1 but set w_1,w_2 to small values in this case.

We first choose the nearest follower in the target lane as the subject's new follower, and estimate its feasibility. The estima-

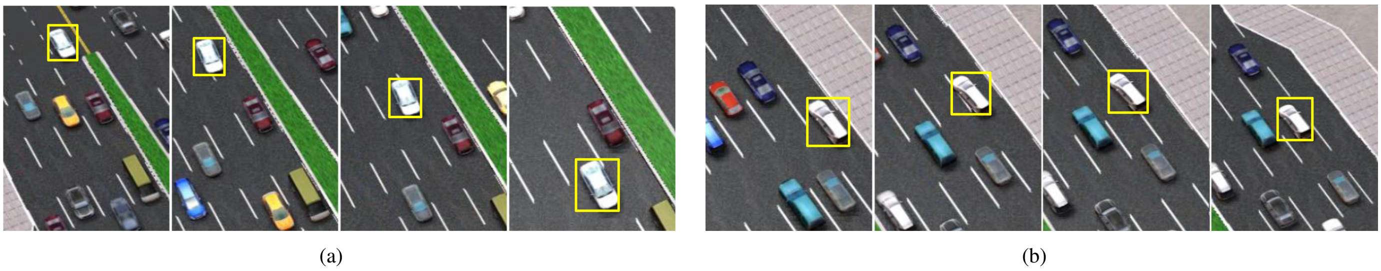


Figure 7: Snapshots of two types of lane-changing motions: (a) The white vehicle's (in yellow frame) free lane change to speedup. (b) The white vehicle's (in yellow frame) imperative lane change due to the reduction of lane number. It needs the new follower's slowing down.

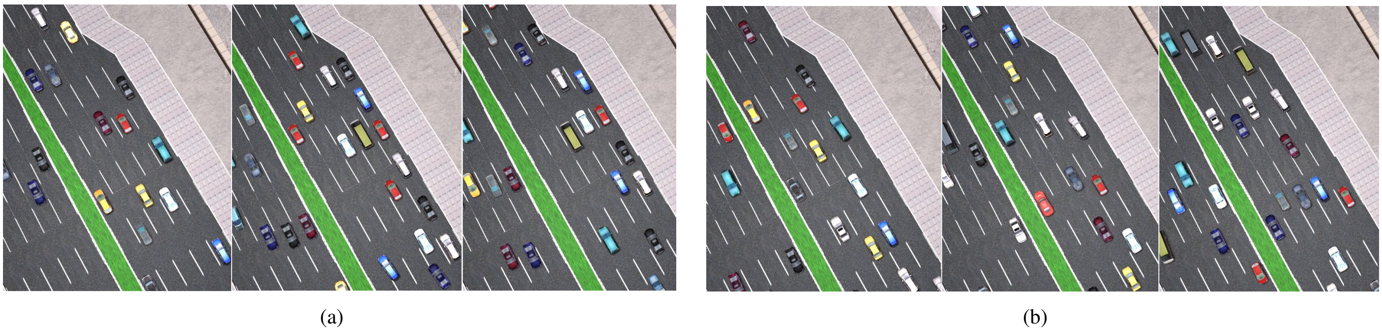


Figure 8: Snapshots of a merging point with imperative lane change disabled (a) and enabled (b): (a) Without imperative lane change, the merge lane can be blocked by those vehicles waiting for large gaps. (b) With imperative lane change, we generate fluent traffic flow.

tion criteria for imperative lane change are briefly described as follows:

$$(a) \quad Dt = \frac{Dv_{nf}}{b_{nf}} \le MAX\_TIME\_LC$$

$$(b) \quad gap'_{ol} \ge gap'_{\min}(s, ol, T_c)$$

$$gap'_{nl} \ge gap'_{\min}(s, nl, T_c) \qquad gap'_{nf} \ge gap'_{\min}(nf, s, T_c)$$

Condition (a) means that we first check the willingness of the chosen new follower to slow down. Dv_{nf} is the speed value that the chosen follower agrees to decrease. It mainly depends on the follower driver's politeness and the urgency of the subject's lane-changing maneuver (expressed as Dis_{fp} - the distance to the forced lane-changing location). b_{nf} is the chosen new follower's comfortable deceleration. If the required cooperating period Dt is larger than MAX_TIME_LC (the maximum lane-changing time selected based on observation), discard this choice and wait for another available and more cooperative follower. Otherwise, go further to check criterion (b).

Condition (b) checks the future gaps based on an assumed lane-changing process (similar with condition b) in section 4.2.1 and Figure 5). In this imaginary lane change, the chosen follower's acceleration would be $-b_{nf}$ for time period Dt. If the involved gaps are safe enough in those checking moments, indicating that this collaborative process for lane change will succeed, the subject will carry on this lane change in the next step and force the follower to decelerate as assumed above. Otherwise, continue to repeat the evaluating process for the next chosen follower until hitting on a workable one.

The implementation of the actual lane change process is similar with the imaginary process in Figure 5: first insert an invisible clone of the subject vehicle into the target lane with zero vehicle length at the starting moment, then update the subject's acceleration according to the new leader, gradually reduce the subject vehicle's length and enlarge the clone's length used in calculation, meanwhile visually adjust the subject's position and orientation towards its clone according to the dynamic turning angle (first increasing then decreasing linearly), finally delete the subject from the old lane and make the clone visible when they coincide. As shown in our supplementary videos, our method could reproduce the visual effect that the lane-changing vehicle modifies its behavior in order to move into the target lane and approaches its new leader naturally.

When the local traffic around the merging point is congested, the imperative lane-changing actions play an important role because the gaps in adjacent lanes are often not large enough for free lane change. Many vehicles in the merge lane may stop and wait if no vehicle's willing to cooperate in target lane (Figure 8(a)). By applying the imperative lane changes, the traffic becomes more fluent (Figure 8(b)). However, during the simulation, we find that when the local traffic is heavy congested, the vehicles could still be blocked in the merge lane. This is mainly caused since they begin to consider imperative lane change when they are quite close to the blocked region, the evaluation processes are more likely to fail. To alleviate this situation, we set an advantage bias in the inner lane and let some vehicles to consider lane change some distance away from the blocked region, leading some vehicles in the merge lane to move into the inner lane a little earlier. By properly integrating the free lane-changing model and imperative lane-changing model, we demonstrated vehicles' natural and adroit lane-changing maneuvers in our traffic animation.

Discussions on our lane-changing model

The main difference between our lane-changing model and the MOBIL model [25] is that ours is a continuous lane-changing process, rather than an instant jump that finishes in a single simulation update. The final visual trajectory is a smooth curve. We have designed a reliable checking procedure to predict the safety of the continuous lane-changing process.

The original Hidas's model [26] proposed a classification of manoeuvres into free, forced and cooperative lane changes. It only describes the final lane-changing action without initial evaluation on driver's lane-changing intention. For forced and cooperative lane change, it first selects a gap, then evaluates the feasibility of new following vehicle to slow down, and finally calculates the merge-acceleration and merge-point to reach the gap. The action is simulated as an instant jump. It also has a complex procedure to check and deal with the conflicting traffic situation (where at the same time the vehicles in the target lane may also need to change into the subject's lane).

Our modification and simplification on Hidas's model can be summarized as follows. Firstly, we abstract the forced and cooperative lane changes into one type, the imperative lane change, since their final visual actions appear identical even through their decision procedures are different. Secondly, we implement the continuous lane-changing actions based on a more reliable checking procedure to ensure the safety during the continuous lane-changing period. Finally, compared to Hidas's abstracted hypothetical roads, much more vehicles are involved in imperative lane changes in our simulating scenarios. Therefore, in order to accelerate this part, we do not calculate the merge-point and merge-acceleration at the beginning of each evaluation process. Instead, we insert an imagery clone of the subject vehicle into the target lane and update the subject's acceleration according to the new leader (Figure 5(a)). Therefore, the subject will automatically accelerate and merge into the target lane by approaching its new leader. To avoid the complicated process of checking and resolving potential conflicts, we force the chosen new follower to wait until the subject has changed its lane. Our modified model, although not very accurate in theory when compared to Hidas's model, is able to produce visually appealing animation with flexible and natural lane-changing actions (see supplementary videos).

5. Road Structures

In our simulator, the urban road network is composed of multi-lane roads with various types of junctions (Figure 1 and 18). We define three basic data structures to represent the multi-lane road, unsignalized junction and signalized junction respectively. Normally each road is related to one junction either unsignalized or signalized, defined as $Junction_{into}$. The detailed lane connections between the incoming and outgoing roads are stored in the junction structure. The connections in the signalized junction are further organized to simulate the signal control of traffic lights.

The basic road structure is a section with the same number of lanes and driving directions. A straight road is simply represented by one line segment with its starting and end point, while the winding one is described as a sequence of connected line segments. Besides the spatial information and the pointer to the related junction, the road structure also contains some information that might be queried, such as speed limit, densities and lane-changing biases for each lane. The bias (related to lane density) will be added to the condition (c) in section 4.2.1 in order to demonstrate the vehicles' ability to take the queue advantage and to better deal with the lane blockage situation as we have described at the end of section 4.2.2. The speed limit indicates the road quality and permitted speed. We compare the vehicle's desired speed with the current road's speed limit, and choose the lower one as the desired speed v_d used in the follow-the-leader model (section 4.1).

In order to quickly respond to the numerous vehicles' queries about the road, the connection and adjacent information, we design a vehicle list for each road to record all the vehicles currently driving on it and make the road to check the answers. The road's vehicle list is sorted based on the vehicles' positions along the path to efficiently handle the subsequent search for neighboring vehicles in the lane-changing process. In addition, we only periodically check the first vehicle in each road's list to find the vehicles that arrive at the road's end and need to be settled according to the junction. Keeping such a list for each road significantly reduces the searching time required in our simulation algorithm. Since the order of the vehicles varies little after one simulation step, we use the bubble sort algorithm to update the list quickly.

The unsignalized junction is an abstract node without concrete spatial structure. According to the driving flow, the junction stores the explicit connections between lanes, for example, the 4th lane of road A is connected to both the 4th lane of road B and the 1st lane of road C. When the vehicle arrives at the end of road, the road checks the structure $Junction_{into}$ to find the right outgoing road and lane. Currently, since our system does not support the routing procedure for the vehicle's destination, if there are more than one outgoing lanes, we randomly select one.

The signalized junction is a structure with concrete paths stored as line segments, such as a typical crossing. In this case, the lane connections will vary according to the repeated signal sets. To simulate the signal control of traffic lights, we first separate the lane connections (defined as *tunnels*) into several groups according to their passing time, add one timer to each group counting its open interval (green period), and then define the changing sequence of these groups (same with the sequence of signal sets). Each *tunnel* keeps a list recording the vehicles in it. We can see the visualized traffic scenario around a typical signalized crossing in the supplementary video.

In each update step, we calculate the vehicles' locomotion using the follow-the-leader model, evaluate and process their lane change actions according to the continuous adaptive lane-changing model, update their positions on the road and sort the vehicle list. For the vehicles inside the *tunnels* of signalized junction, we just update and advance their position along the *tunnel* paths. After checking the first vehicle in each vehicle list, we further settle those vehicles that arrive at the end of the road using the linking information stored in $Junction_{into}$ and advance those vehicles that finish their *tunnel* paths:

1) If there is no junction at the road's end, just discard the vehicle since we assume it has finished the driving task on our focus network.

2) If $Junction_{into}$ is an unsignalized junction, insert the vehicle into the connected road's vehicle list according to the lane connection stored in the junction, modify the vehicle's parameters including lane index, road index and position along the road path, and finally delete it from the old road's vehicle list.

3) If $Junction_{into}$ is a signalized junction, find the *tunnel* for the vehicle's lane and throw the vehicle into this *tunnel's* vehicle list, modify the related parameters and delete it from the old road's vehicle list.

4) If the vehicle in the signalized junction reaches the end of its *tunnel* path, push it into the vehicle list of the outgoing road related to this *tunnel*, modify the related parameters and delete it from the *tunnel's* vehicle list.

For the border roads without incoming roads, we continuously insert vehicles into them based on the in-flow quantity. Users can set the initial traffic density and adjust the in-flow quantity during simulation to obtain sparse or dense traffic scenarios. It is also flexible for traffic analysts to modify the control signal's timer values to regulate the traffic.

6. Experimental Results

We have performed our experimental tests on a desktop PC equipped with Intel(R) Core(TM)2 Duo CPU E7500@2.93GHz, 4GB main memory (3.25GB available) and an NVIDIA GeForce GTX260 graphics card.

6.1. Performance Test

We have tested the update rate of our method on a long road with four lanes. The initial traffic density on the road is about $0.05 vehicles/m/lane$. The vehicle gap in a single lane is around $10 - 35m$. The gap between two vehicles in adjacent lanes is about $2.5 - 7.5m$. During the experiment, we maintained a constant number of vehicles on the road by inserting a new vehicle whenever there's a vehicle that finished the road. The vehicles' parameters were randomly set according to the normal range defined in Table1. We applied our continuous lane-changing model and chose the related variables as follows: $\Delta a_{thr} \in [0.08, 0.12]m/s^2$, $pf \in [0.1, 0.4]$, $MAX\_TIME\_LC \in [1.8, 3.7]s$, $Dv_{nf} \in [2.8, 4.8]m/s$. The time step Δt used in the update is $0.04s$, and the vehicle will stop the lane-changing estimating process for $5 - 20$ seconds (waiting interval) when it has just changed the lane.

We first chose the number of testing vehicles, and then ran our program for 120 seconds, recording the update counts and the number of vehicles performing lane-changing actions in 20 randomly sampled updates. We calculated the average simulation time of these vehicles for each frame and showed the result in Figure 9. The maximum number of vehicles we can simulate in real time is around 40000. According to the statistics of our experiments with the above settings, the success rate of lane-changing estimating process is about 20%, the average percentage of vehicles performing lane-changing actions in each frame is 5.53%, in which, the imperative behaviors occupy 23.61%. This means in each update of 100 vehicles, there are about five vehicles changing their lanes and one of them needs its neighbor's cooperation.

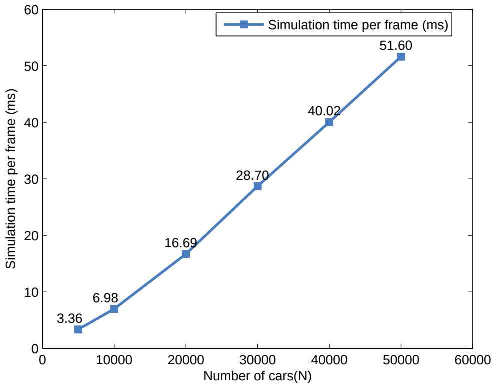


Figure 9: The average simulation time per frame for N vehicles.

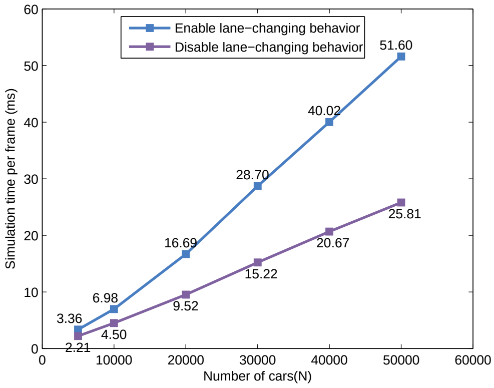


Figure 10: The simulation time per frame when enabling (blue line) and disabling (purple line) lane-changing behaviors.

We have introduced a continuous lane-changing model to our microsimulation scheme, resulting in an additional time expense. We have further quantified this expense by testing the computation time of the lane-changing model. The result is shown in Figure 10. It compares the simulation time used per frame when enabling and disabling the lane-changing behaviors respectively in different simulation scales. The computing time proportion relevant to lane-changing roughly increases with the simulation scale, from 36% to 51% out of the total simulation time.

In our algorithm, the estimating procedure of possible lane changes is a comparatively time-consuming part. Raising the

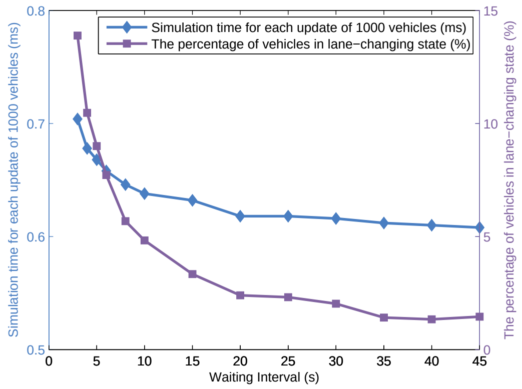
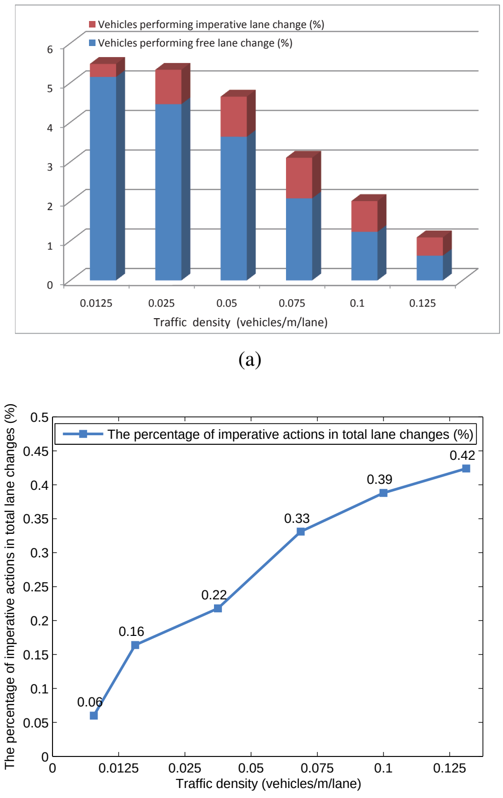


Figure 11: The influence of the lane-changing estimating frequency (set by the waiting interval for next estimating procedure) on both the simulation time for each update and the final proportion of vehicles performing lane changes.



(a)

(b)

Figure 12: (a) The influence of traffic density on final lane-changing actions. (b) The influence of traffic density on imperative lane-changing proportion in total lane changes.

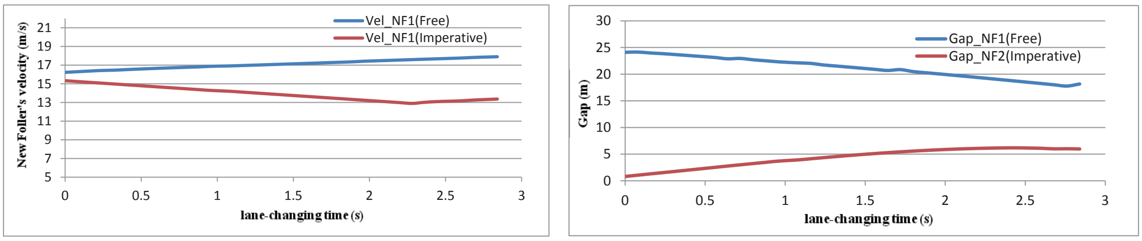
frequency of the vehicle's estimating process increases the simulation time for each update and also the number of successful lane-changing actions. In the second experiment, we have tested the impact of the frequency of the estimating process on the simulation time and final lane-changing actions in each update. The frequency was set by the vehicle's waiting interval between two estimating processes. Most experimental parameters remained unchanged except that in this case, we have tested 1000 vehicles for 180 seconds. In Figure 11, we show that both the simulation time for each update and the number of vehicles performing lane-changing actions decrease when we increase the waiting interval (which means we lower the lane-changing estimating frequency).

In the subsequent experiment, we varied the traffic density on a road with four lanes and 40km long to see its influence on the lane-changing actions using our continuous lane change model. The waiting intervals were set to be 10 seconds for all vehicles to eliminate their impact on lane changes. From Figure 12(a), we can see that when we increase the traffic density, the percentage of vehicles performing lane-changing actions decreases since more lane-changing estimating processes would fail. Among these lane-changing actions, the proportion of imperative lane changes increases when increasing traffic density (as shown in Figure 12(b)). This phenomenon can be observed in real-life situations: when the road becomes congested, there will be less lane-changing behaviors and more of them would require other vehicles' cooperation.

The success rate of the lane-changing estimating process is more or less influenced by some other factors besides the traffic density, such as the vehicle's parameters and driver's preference (Δa_{thr}, pf, MAX_TIME_LC, Dv_{nf}). Users can adjust these parameter ranges to obtain higher or lower lane-changing possibility.

There are mainly two differences between the free lane-changing action and the imperative lane-changing action. The first difference is that the imperative type will require the new follower in the target lane to slow down first. The second one is that the gap available for the subject vehicle to insert into in the imperative action is usually smaller than the free situation. We have verified these differences by tracking the new follower's velocity variance and its gap to the subject vehicle during the lane-changing process. The results are shown in Figure 13. From Figure 13(a), we can see that the subject's cut-in action has little effect on the new follower in the free lane change (blue line) since their gap is very large. However, for the imperative lane change (red line), the new follower first slows down to enlarge their gap. From Figure 13(b), we can see that the gap in the imperative action is smaller than the free case and it will increase to allow for the subject's cut-in.

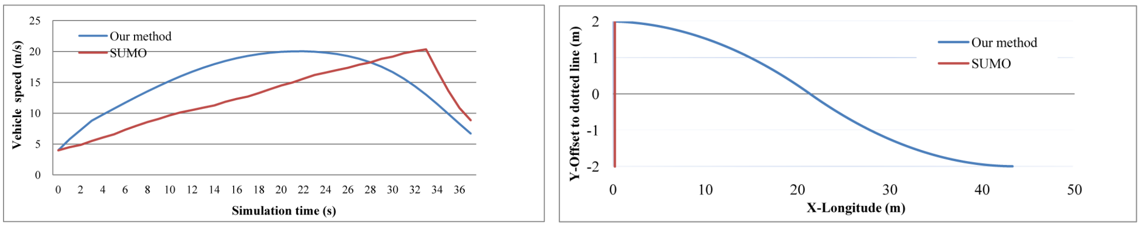
For comparison, we have recorded the vehicle's speed on a straight road by using our method and SUMO [23] separately with the same initial conditions (including the same neighboring gaps and the vehicles' equal driving abilities). The road is 500m long and is blocked at the end. The traced vehicle is initially set around the beginning of the road and will stop before the blockage. The result shows that our extended driving model can produce more realistic motion by accelerating and braking more smoothly than SUMO (see Figure 14(a)). We have also traced a randomly selected vehicle's lane-changing trajectory on a multi-lane road in our simulation and SUMO [23] respectively. Figure 14(b) illustrates that our lane-changing path is a smooth curve while the lane-changing used in SUMO is an instant jump finished in a single simulation step.



(a)

(b)

Figure 13: (a) A comparison on the new follower's speed variance of free lane change in blue vs. imperative lane change in red. (b) A comparison on the new follower's front gap variance of free lane change in blue vs. imperative lane change in red.



(a)

(b)

Figure 14: (a) A comparison on accelerating strategy of SUMO in red vs. our method in blue. The vehicle first accelerated for its desired velocity and then decelerated when approaching the blockage at the end of the road. (b) A comparison on lane-changing trajectory of SUMO in red (an instant jump) vs. our method in blue (a smooth curve).

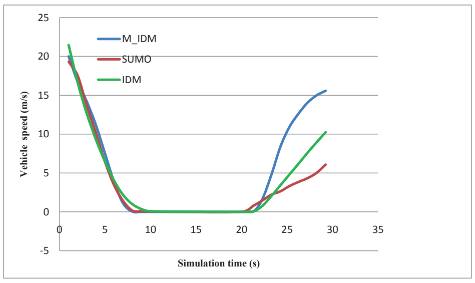


Figure 15: A comparison on velocity variance between several follow-the-leader models in a scenario when the vehicle first stopped before a crossing and then accelerated when the traffic light turned green: the original IDM model [16] in green, our modified model in blue and Sumo [23] in red.

Moreover, we have recorded the vehicle's velocity in a crossing scenario for a period using the original IDM model, our modified model and SUMO separately. The vehicle first decelerated and stopped before the crossing, and later accelerated when the traffic light turned green. The comparison in Figure 15 shows that the vehicle will accelerate faster when using our relaxed deceleration term (modified from the original IDM model). Our modification can produce a smoother restarting process in our animation.

The diversity of the drivers' behaviors is determined by the parameter variation. Table 2 is a summary of the effect on the final simulation of the basic vehicle parameters. Additionally, we have compared the simulation results using different parameter variations (Figure 16 and more comparisons in the accompanying demo). To investigate a parameter's effect, we have performed simulations by changing its variation range from ±5% to ±40% (while keeping other parameters varying within ±5%). The experiment demonstrates that larger parameter variation will result in more diverse actions in the final simulation.

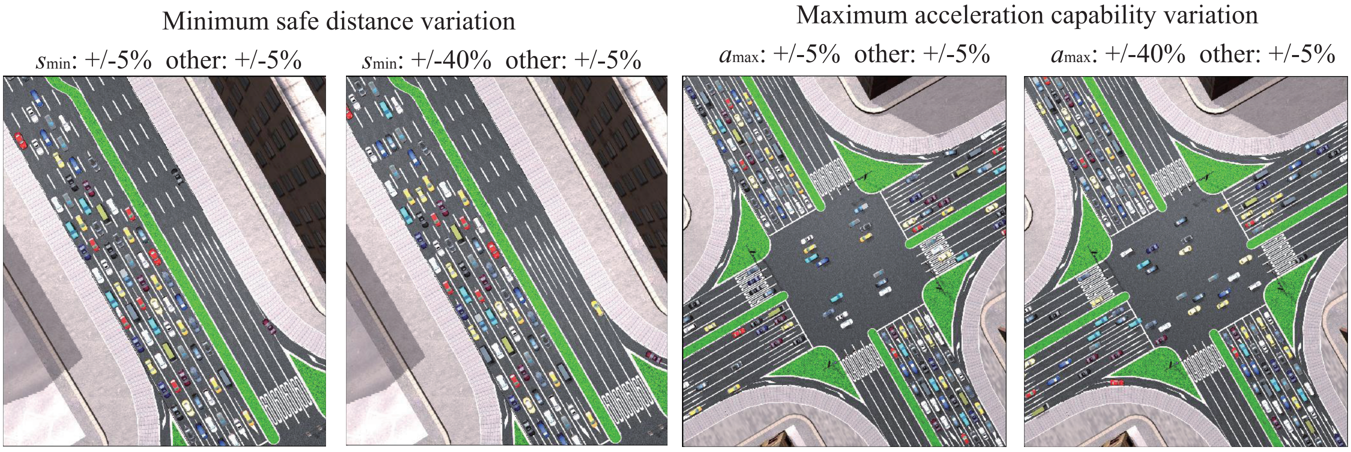
6.2. *Simulation Results*

We have built a typical urban road network with signalized crossings, merging junctions and elevated roads, and visualize the various traffic scenarios using our agent-based system with continuous lane-changing implementation (see Figure 1, 7 and 18). Our system can motivate each vehicle intelligently as if a real driver was in it. You can observe the various driving styles and natural lane-changing behaviors on diverse roads and junctions in the supplementary videos. All the presented animation results are rendered in real time by using the open source Horde3d Engine [28].

Compared with other microscopic simulators like SUMO [23], our approach is computationally less efficient due to the continuous lane-changing interactions for realistic 3D animation purpose. However, the performance is still competitive and is able to simulate 40000 vehicles in real time. In addition, we have covered a rich variety of vehicle types, driving styles and urban road topologies, integrated more intelligent lane-changing behaviors and overcome the limitations in other 3D traffic visualizations ([10] [24]). More detailed comparisons are shown in Figure 17.

7. Conclusion

We have presented a new microscopic system to generate detailed 3D traffic animations on complex road networks with diverse junctions by using an improved intelligent driver model together with a continuous lane-changing technique. We


Minimum safe distance variation
$s_{\min}$: +/-5% other: +/-5%
$s_{\min}$: +/-40% other: +/-5%
Maximum acceleration capability variation
$a_{\max}$: +/-5% other: +/-5%
$a_{\max}$: +/-40% other: +/-5%

(a) (b)

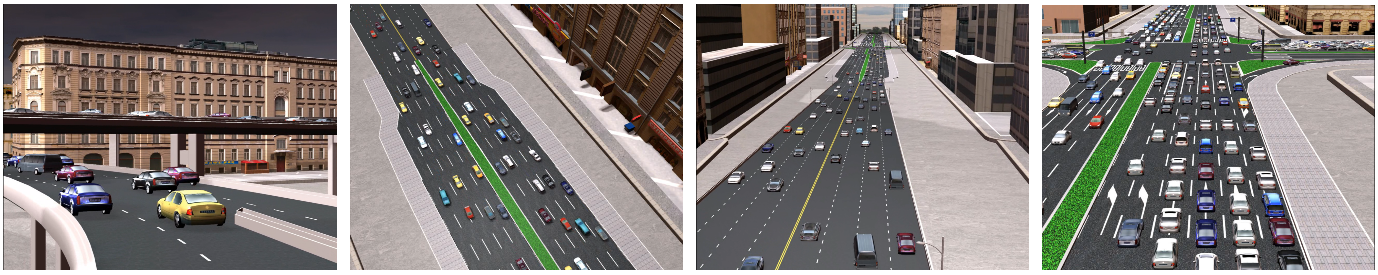
Figure 16: The effect on the final result caused by different parameter variations (see the vehicles in red box): (a) larger variation of s_0 resulted in larger static gap variance. (b) larger variation of $a_{\max}$ resulted in more diverse accelerating reactions.

Table 2: A summary of vehicle parameters' effects on simulation results.

Parameter	Effect on the simulation result
v_d	The speed the driver desires to reach. Larger values result in faster driving motions on roads.
$s_{\min}$	It can be viewed as the safe distance in traffic jam.
$a_{\max}$	The vehicle with a larger value accelerates more quickly especially when restarting from a static state.
T	It determines the clearance (velocity-dependent) the driver prefers to keep for relax driving.
b_{com}	It has more intuitive meaning in the lane-changing model, see condition (a) in Section 4.2.1 and 4.2.2. The vehicle with a larger value reacts more supportively for others to cut into the position in front of it.

simulation approach	Visual result	Data driven	Model category	Motivation of lane change	Cooperative cut-in	Implementation of lane change	Vehicle types, Driving styles	Road structures	Performance (ten thousand scale)
Sumo	2D	No	Microscopic	Pre-defined or speedup	No	An instant jump	Rich	Rich	Real-time
CTS	3D	No	Macroscopic	Pre-defined or speedup	No	A curve (almost fixed)	Limited	Limited	100x Faster than real-time
VTR	3D	Yes	Microscopic (multi-robot planning)	Pre-defined by planning	No	A fixed curve on roadmap	Limited	Limited	Off-line
Our method	3D	No	Microscopic	Speedup or conditionally required by driver	Yes	A dynamic curve defined by vehicles' speeds	Rich	Rich	Real-time

Figure 17: A comparison on several traffic simulation methods: Sumo (a representative of common traffic simulators), CTS (a continuum method [Sewall et al. 2010b]), VTR(a reconstruction framework [Sewall et al. 2010a]) and our method.



(a) (b) (c) (d)

Figure 18: Some images captured in the road network: (a) A merging/weaving part. (b) A junction with lane number reduction. (c) Driving down from the elevated road. (d) Approaching a signalized crossing.

demonstrate how our method can be applied to produce believable urban traffic animations with more flexible driving styles and realistic lane-changing behaviors. Since our method is agent-based, we can deal with each vehicle individually and motivate it as an autonomous particle that can play decision-making roles and take corresponding actions to drive as fast as possible, and meanwhile, keep safe from possible collisions and pass the junctions naturally.

In our current implementation, the vehicle has no specific destination. The driving direction is determined by the road it follows and the random selections when approaching the junctions. To simulate a driver's real actions to reach his specific destination, an effective route searching part can be added for the simulator in future. Furthermore, the current simulation is limited to private vehicles and does not take public transportation tools into consideration, such as BRT (Bus Rapid Transit) which has its own specific driving lane, and motorcycle characterized with a freer driving style. Lastly, we can add pedestrians into our framework to simulate more appealing scenarios in future.

Acknowledgements

This work was supported by Zhejiang Provincial Natural Science Foundation of China (Grant no. Z1110154), the China 973 program (Grant no. 2009CB320801), the National Natural Science Foundation of China (Grant nos. 60933007 and 60833007), and NSFC-MSRA Joint Funding (Grant no. 60970159).

References

[1] D. Helbing, Traffic and related self-driven many-particle systems, Reviews of Modern Physics 73 (4) (2001) 1067–1141.

[2] M. J. Lighthill, G. B. Whitham, On kinematic waves. ii. a theory of traffic flow on long crowded roads, in: Proceedings of the Royal Society of London. Series A, Vol. 229, Mathematical and Physical Sciences (1934-1990), 1955, pp. 317–345.

[3] G. F. Newell, A simplified theory of kinematic waves in highway traffic, part 1: General theory, Transportation Research Part B 27 (4) (1993) 281–187.

[4] C. F. Daganzo, The cell transmission model, part ii: Network traffic, Transportation Research Part B 29 (2) (1995) 79–93.

[5] J. P. Lebacque, A finite acceleration scheme for first order macroscopic traffic flow models, in: Proceedings of the 8th IFAC symposium on transportation systems, Vol. 2, International Federation of Automatic Control, Chania, Greece, 1997, pp. 815–820.

[6] H. J. Payne, Models of freeway traffic and control, Mathematical Models of Public Systems 1 (1) (1971) 51–61.

[7] G. B. Whitham, Linear and nonlinear waves, Wiley, New York, 1974.

[8] A. Aw, M. Rascle, Resurrection of second order models of traffic flow, SIAM Journal of Applied Math 60 (3) (2000) 916–938.

[9] H. M. Zhang, A non-equilibrium traffic model devoid of gas-like behavior, Transportation Research Part B 36 (3) (2002) 275–290.

[10] J. Sewall, D. Wilkie, P. Merrell, M. C. Lin, Continuum traffic simulation, Computer Graphics Forum 29 (2) (2010) 439–448.

[11] L. A. Pipes, An operational analysis of traffic dynamics, Applied Physics 24 (3) (1953) 274–281.

[12] P. E. Chandler, R. Herman, E. Montroll, Traffic dynamics: Studies in car following, Operations Research 6 (2) (1958) 165–184.

[13] D. L. Gerlough, Simulation of freeway traffic on a general-purpose discrete variable computer, PhD thesis, UCLA (1955).

[14] D. C. Gazis, R. Herman, R. W. Rothery, Non-linear follow-the-leader models of traffic flow, Operations Research 9 (4) (1961) 545–567.

[15] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, Y. Sugiyama, Dynamic model of traffic congestion and numerical simulation, Physical Review E 51 (2) (1995) 1035–1042.

[16] M. Treiber, D. Helbing, Microsimulations of freeway traffic including control measures, Automatisierungstechnik 49 (2001) 478–484.

[17] A. Kesting, M. Treiber, D. Helbing, Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity, Philosophical Transactions of the Royal Society A 368 (1928) (2010) 4585–4605.

[18] K. Nagel, M. Schreckenberg, A cellular automaton model for freeway traffic, Journal de Physique I 2 (12) (1992) 2221–2229.

[19] Corsim, Corridor simulation (corsim / tsis), http://ops.fhwa.dot.gov/trafficanalysistools/corsim.htm (Dec. 2010).

[20] Mitsim, Mitsimlab open source (open source version), http://mit.edu/its/mitsimlab.html (Nov. 2002).

[21] G. Cameron, B. Wylie, D. McArthur, Paramics: Moving vehicles on the connection machine, in: Proceedings of the 1994 ACM/IEEE Conference on Supercomputing, IEEE Computer Society Press Los Alamitos, CA, USA, 1994, pp. 291–300.

[22] K. A. Redmill, U. Ozguner, Vatsim: A vehicle and traffic simulator, in: IEEE 2nd International Conference on Intelligent Transportation Systems, Intelligent Transportation Systems, 2001, pp. 656–661.

[23] Sumo, Simulation of urban mobility (open source version), http://sumo.sourceforge.net/ (Oct. 2009).

[24] J. Sewall, J. van den Berg, M. C. Lin, D. Manocha, Virtualized traffic: Reconstructing traffic flows from discrete spatio-temporal data, IEEE Transactions on Visualization and Computer Graphics 17 (1) (2010) 26–37.

[25] A. Kesting, M. Treiber, D. Helbing, General lane-changing model mobil for car-following models, Transportation Research Record 1999 (10) (2007) 86–94.

[26] P. Hidas, Modelling vehicle interactions in microscopic traffic simulation of merging and weaving, Transportation Research Part C: Emerging Technologies 13 (1) (2005) 37–62.

[27] A. H. M. Treiber, D. Helbing, Congested traffic states in empirical observations and microscopic simulations, Physical Review E 62 (2) (2000) 1805–1824.

[28] Horde3d, Next-generation graphics engine (open source), http://www.horde3d.org/ (Oct. 2009).