A High Precision and Efficient Time-to-Collision Algorithm for Collision Warning Based V2X Applications

Chao Qu
School of computer science and software engineering
East China Normal University
Shanghai, China
e-mail: 51164500152@stu.ecnu.edu.cn

Wen-Yuan Qi
Connected Driving Experience Research Group (China Science Lab)
General Motors Corporation
Shanghai, China
e-mail: jimmy.qi@gm.com

*Ping Wu
computer science and software engineering
East China Normal University
Shanghai, China
e-mail: pwu@cc.ecnu.edu.cn

***Abstract*—Time-to-Collision(TTC) algorithm is the key component for collision warning based Vehicle-to-Everything applications. In this paper, a GPS based intersection model is firstly introduced based on the relative position and distance between vehicles or traffic participants. This model avoids the extra computing time required to create a Cartesian coordinate system. Secondly, a novel rectangle model is proposed based on our GPS model in order to represent the shape of the vehicle more accurately. It can also be used to identify different collision situations at the intersection more precisely. Through which, the experiments show that TTC can be calculated more accurately than other shape models, and more effectively in reducing the consumption of calculation for non-collision situations.**

Keywords-V2X, TTC; collision warning algorithm; GPS

I. INTRODUCTION

Approximately 1.24 million people around the world die and between 20 and 50 million withstand non-fatal injuries on roads every year. And road accidents are predicted to increase by 65% and become the fifth major cause of death by 2030 [1] and about 30% of all road accidents occur at intersection in China. To reduce high risks of the above situations, inter-vehicle communication technologies including DSRC and cellular-based data transmission services are developed rapidly [2]. With the help of these high efficiency communication technologies, high-precision GPS and vehicle information can be transmitted between vehicles at minimal latency. Many active V2V(vehicle-to-vehicle) safety applications could be realized for achieving better driving experience and driving safety [3]. Among the collision warning based V2V applications, Time-to-Collision (TTC) algorithm is the key and basic component of vehicle collision detection system since it is aimed at notifying drivers of upcoming threats and potential collisions. The idea of time-to-collision (TTC) was first defined as “the time required for two vehicles to collide if they continue at their present speed and on the same path.” by Hayward [4]. In this paper, we proposed a high-precision TTC algorithm for more accurate collision detection. This could also be leveraged by V2B (vehicle-to-bike) and V2P(vehicle-to-pedestrian) safety applications [5]. In the following sections, related work will be described in Section II. Our proposed novel TTC algorithm will be detailed explained in Section III. Section IV will illustrate advanced topics of TTC algorithm filtering methods followed by some experimental results that is better than other related models, and final Section V will be the conclusion and future work.

II. RELATED WORK

Traditional TTC algorithms in previous researches can be divided into three classes according to the model used to describe traffic participants: (1) point model (2) circle model (3) rectangle model. Point model is mainly used in industry, the motor vehicles, bicycles, or people are all considered as points in collision model [6] [7] [8]. This model caused some passed by situations could be wrongly judged to a collision. It will also give wrong TTC when the included angle between two traffic participants is small. As to the circle model [9][10], although the circle method can solve the problem exists in the point model to some extent, it is not accurate enough, and it can also produce false alarm. For example, when two vehicles pass relatively at a close distance, the circle model will misjudge the situation as collision too. The rectangular model is the closest approach model to actual vehicle shape in TTC algorithm [9] [10] [11] [12]. But in the previous study, both general rectangular model and grid-based space-time synchronization algorithm [13] were mainly used the time simulation method to calculate TTC [9], that is to calculate whether the vehicle position will overlap at the next moment. However, the time simulation method will consume huge computing resources, and it also has the disadvantage of not providing the exact TTC as well. For those methods that do not use simulation, there is also a problem that does not take into account all the crash situations which make the algorithm not accurate enough [10] [11] [12].

978-1-5386-7370-6/18/$31.00 ©2018 IEEE

III. Proposed TTC Algorithm

In order to solve the above problems, this paper presents a new rectangle model which can calculate the precise TTC in an efficient way. The methodology is mainly composed of three parts: First part introduces a GPS model based intersection model to calculate the related position of target vehicle. Second part is a novel rectangle model to classify and identify various collision situations. Third part presents a procedure to minimize the consumption of the computation resources (this part will be detailed discussed in Section IV).

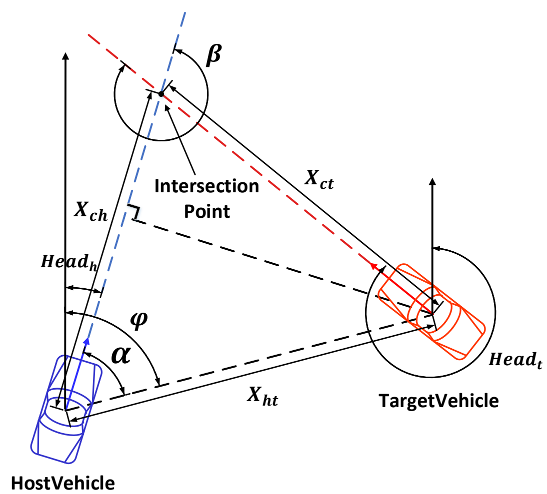


Figure 1. Schematic of collision scenario.

A. GPS Based Intersection Model

In order to avoid the extra computing time required to create a Cartesian coordinate system. A GPS based intersection model is firstly introduced based on the relative position and distance between vehicles. The GPS model is based on the assumptions as follows, which has limited affection to the points within a short distance. a) The earth is a standard sphere. b) $\pi = 3.1415926$. c) The radius of Earth $R = 6378137$m. d) Longitude line (Earth equatorial circumference) $L = 2 \cdot \pi \cdot R = 40075016$m. e) Each degree latitude changes, Longitude keeps same, distance will change as $LatL = L / 360 = 111319$m.

Generally, GPS information of host vehicle is retrieved by built-in vehicle GPS system or external high-precision GPS device. Information of Latitude Lat_h (degree), Longitude Lon_h (degree), speed V_h (meter/second) and heading $Head_h$ (degree) are necessary. On the other hand, the target vehicle GPS information is transmitted by DSRC or LTE network, including Latitude Lat_t(degree), longitude Lon_t (degree), speed V_t (meters/second) and heading $Head_t$(degree).

Furthermore, target vehicle orientation related to the host vehicle is another basic metric for TTC calculation in most of collision warning based V2X applications. The target orientation angle α could be easily calculated by GPS information of host and target vehicle, which will be described in the following section.

1) Target orientation angle

In order to calculate the relative position of the host vehicle and the target vehicle, we use the target orientation angle to describe the position of the target vehicle. The target orientation angle is defined as the angle between the GPS position of the target vehicle and the driving direction of the host vehicle. In order to calculate the target orientation angle, we first need to calculate the geographic position angle of the target vehicle relative to the host vehicle. The geographic position angle of the target vehicle in figure 1 is φ. It is defined as the angle between target vehicle GPS position and the longitude line. Its value can be calculated using the following formula:

$$\varphi = arc\,tan\left(\frac{LatL \cdot cos\,Lat_t \cdot (Lon_t - Lon_h)}{LatL \cdot (Lat_t - Lat_h)}\right)$$
$$= arc\,tan\left(\frac{cos\,Lat_t \cdot (Lon_t - Lon_h)}{Lat_t - Lat_h}\right) \quad (1)$$

Note that the vertical direction of each vehicle on Fig.1 is not the heading of vehicles, but the geographic north axis. Once φ is calculated, the target vehicle orientation angle α which between the host vehicle's heading and the connection of host vehicle and target vehicle can be calculated by the equation below.

$$\alpha = \varphi - Head_h \quad (1)$$

The delta heading, also called heading deference angle β which is the angle between two vehicle's heading can be formulated as follows:

$$\beta = Head_t - Head_h \quad (3)$$

The target vehicle orientation angle α and heading between two vehicles β calculated above may greater than 360° or less than 0°, so it need to be modified between 0° and 360°.

2) Collision scenario

Fig.1 shows a typical GPS based collision scenario for calculating TTC in this paper. It can be seen that the trajectories of two vehicles will cross at the intersections when the they continue to travel according to the current trajectory. The geographic position angle φ, the target orientation angle α and the heading deference angle β are all indicated in figure 1. In this figure, the vehicle on the left side is the host vehicle, and the right vehicle is the target vehicle. When the vehicle receives information from other vehicles through LTE or DSRC transmission, first calculate the distance between the two vehicles X_{ht} can be calculate by the equations below:

$$X_{ht} = 2 \cdot R \cdot arcsin\sqrt{sin^2\frac{\Delta Lat}{2} + cos\,Lat_h \cdot cos\,Lat_t \cdot sin^2\frac{\Delta Longt}{2}} \quad (4)$$

$$\text{here,} \begin{cases} \Delta Lat = Lat_t - Lat_h \\ \Delta Longt = Lon_t - Lon_h \end{cases} \quad (5)$$

Besides, distances of both vehicle's current location to the intersection point will be used to identifying different collision situations, which can be calculated using geometric methods. Where X_{ch} represents the distance between intersection point and current host vehicle GPS location. Similarly, X_{ct} denotes the distance between intersection point and current target vehicle GPS location.

$$X_{ch} = X_{ht} \cdot (\cos\alpha - \sin\alpha / \tan\beta) \tag{6}$$

$$X_{ct} = -X_{ht} \cdot \sin\alpha / \sin\beta \tag{7}$$

B. Rectangle Model for Identifying Collision Situations

This section introduces a novel rectangular collision model which considers vehicle as rectangle, since rectangular is the closest shape to the real shape of the vehicle. In actual cases, the collision between two cars completely parallel or vertical is nonexistent, so four corners of the rectangular vehicle model

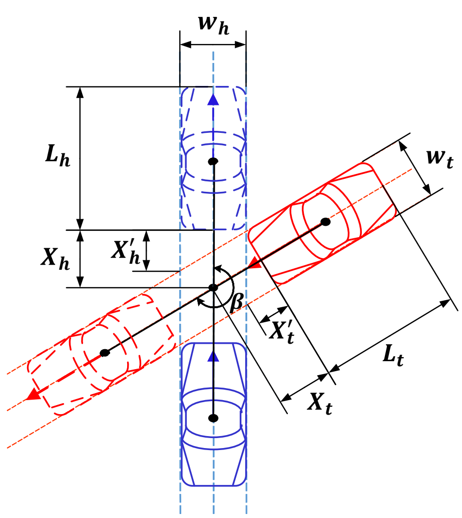


Figure 2. Vehicles just reach and just exit the intersection.

will collide with four sides once collision occurs. According to the angle β between two vehicle headings, the collision situations can be divided into two categories, category 1 contains 4 cases when $\beta \in (90°, 270°)$ and category 2 contains 6 cases when $\beta \in (0°, 90°) \cup (270°, 360°)$. To identify precise difference of each collision situations, key timing of potential collision will be newly introduced in this section.

1) Intersection area

The model where two vehicles approaching the intersection area is shown in the Fig. 2. The red vehicle represents the target vehicle, and the blue vehicle represents the host vehicle. Vehicle GPS position are located at the center of the vehicle. The blue dotted line denotes the vehicle's driving trajectory, and the distance between two blue dotted lines is the width of the host vehicle W_h. In the same way, the red dotted line denotes the trajectory of the target vehicle, and the distance between the two red lines is equal to the target vehicle width W_t. The solid line vehicle indicates the position at which the vehicle just entered the intersection, and the dotted line car indicates where the car just left the intersection.

$$\theta = \begin{cases} \beta & ,\beta \in [0, 90°) \\ 180 - \beta & ,\beta \in [90°, 180°) \\ \beta - 180 & ,\beta \in [180°, 270°) \\ 360 - \beta & ,\beta \in [270°, 360°) \end{cases} \tag{8}$$

Fig. 2 shows the distances required to calculate key timing, where L_t and L_h are the length of the target vehicle and the host vehicle respectively. X_t, X_h are the distance between the front of two vehicles and the intersection point, calculated by (9) and (10).

$$X_t = \left(\frac{W_t}{2}\cos\theta + \frac{W_h}{2}\right) / \sin\theta \tag{9}$$

$$X_h = \left(\frac{W_h}{2}\cos\theta + \frac{W_t}{2}\right) / \sin\theta \tag{10}$$

Furthermore, X'_t denotes the distance between the left-end point of the target vehicle's front side and the right track of the host vehicle along the target vehicle's left track marked in the above Fig. 2. X'_t is only determined by the angle between the driving directions of two vehicles and the width of target vehicle. According to the same principle, X'_h changes according to the host vehicle width W_h.

$$X'_t = W_t / \tan\theta \tag{11}$$

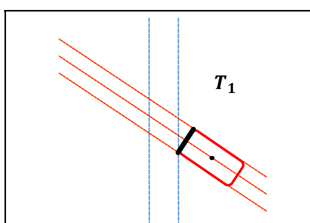
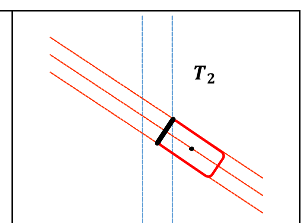
$$X'_h = W_h / \tan\theta \tag{12}$$

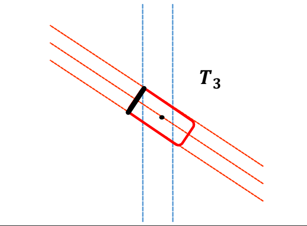
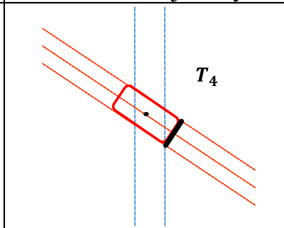
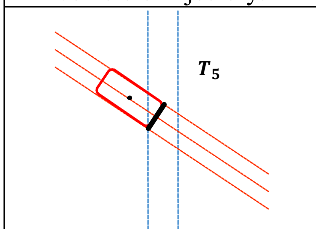
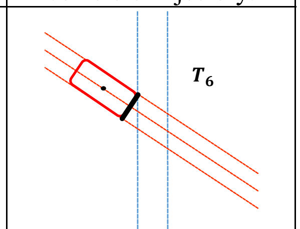
Since the model is symmetrical by the intersection point, the corresponding distance when the vehicle exits the intersection equals to the distances calculated when the vehicle enters the intersection.

2) Definition of key timing

In order to calculate the accurate TTC for each collision situation, it is necessary to determine the specific collision type case by case. We can determine which side of a particular vehicle will be hit by calculating the time required for vehicle arriving key points near the intersection refer to Table I.

TABLE I. 6 KEY POINT WHEN VEHICLE ENTERED THE INTERSECTION

	
(a) The vehicle's front side	(b) The vehicle's front side

just reached the other vehicle's trajectory.	just entered the other vehicle's trajectory.
	
(c) The vehicle's front side just exited the other vehicle's trajectory.	(d) The vehicle's rear side just reached the other vehicle's trajectory.
	
(e) The vehicle's rear side just about to leave the other vehicle's trajectory.	(f) The vehicle's rear side just exited the other vehicle's trajectory.

$$\begin{cases} T_1 = (X_c - X - L/2)\,/\,V \\ T_2 = (X_c - X + X' - L/2)\,/\,V \\ T_3 = (X_c + X - L/2)\,/\,V \\ T_4 = (X_c - X + L/2)\,/\,V \\ T_5 = (X_c + X - X' + L/2)\,/\,V \\ T_6 = (X_c + X + L/2)\,/\,V \end{cases} \quad (13)$$

X, X' denotes the distance illustrated in Fig 2. As the concept of target vehicle and host vehicle is relative, the definitions are applicable to both vehicles.

T_1, T_2, T_3, T_4, T_5 and T_6 represent the key time taken by the vehicle arriving at the position. As is shown in TABLE I respectively based on the current vehicles status, including current location, current speed and vehicle heading with both vehicles.

In the following of this paper, T_t will be used to represent the time required for the target vehicle to reach key points, and T_h to represent the time required for the host vehicle to reach key points.

C. TTC Calculation

1) Collision situations when $\beta \in (90°, 270°)$

In this case, two sides of each vehicle have the possibility to be collided (the front side of the other vehicle and left or right side of the vehicle. Collision situation can be divided into four categories according to the side to be collided as shown in TABLE II (case $\beta \in (90°, 180°)$). TTC could be calculated according to key time T in each case (Equation 14-17). Due to the symmetry of the model (target vehicle approaching from the left), the time of collision in case $\beta \in (180°, 270°)$is calculated in the same way as TABLE II shows.

$$TTC_1 = T_{t1} \quad (14)$$

$$\mathrm{TTC}_2 = \mathrm{T}_{h1} \quad (15)$$

$$\mathrm{TTC}_3 = \left(\frac{W_t}{\sin\theta} + T_{h2}V_h + \frac{T_{t1}V_t}{\cos\theta}\right) / \left(\frac{V_t}{\cos\theta} + V_h\right) \quad (16)$$

$$\mathrm{TTC}_4 = \left(\frac{W_h}{\sin\theta} + T_{t2}V_t + \frac{T_{h1}V_h}{\cos\theta}\right) / \left(\frac{V_h}{\cos\theta} + V_t\right) \quad (17)$$

2) Collision situations when $\beta \in (0°, 90°) \cup (270°, 360°)$

Six specific types of collision will be covered in this case as shown in TABLE III. Three sides of a vehicle can be collided: the front side, the rear side and one of the left or right side. Different from situations that β is greater than 90 degrees (β ϵ (90°,270°)), two additional cases (6) and (10) are covered. In these two cases, vehicles may collide on their rear sides with key timing discussed in (d) and (e) from TABLE I. Total six kinds of collisions that may occur when the target vehicle approaches the target point from the right side of the vehicle at less than 90 degrees in TABLE III. TTC could be precisely calculated as follows:

$$\mathrm{TTC}_5 = \mathrm{T}_{t1} \quad (18)$$

$$\mathrm{TTC}_6 = (T_{h5}V_h - T_{t2}V_t\cos\theta)\,/\,(V_h - V_t\cos\theta) \quad (19)$$

$$\mathrm{TTC}_7 = (T_{h1}V_h - T_{t4}V_t\cos\theta)\,/\,(V_h - V_t\cos\theta) \quad (20)$$

$$\mathrm{TTC}_8 = \left(T_{h4}V_h - \frac{T_{t1}V_t}{\cos\theta}\right) / \left(V_h - \frac{V_t}{\cos\theta}\right) \quad (21)$$

$$\mathrm{TTC}_9 = \mathrm{T}_{h1} \quad (22)$$

$$\mathrm{TTC}_{10} = \left(T_{h2}V_h - \frac{T_{t5}V_t}{\cos\theta}\right) / \left(V_h - \frac{V_t}{\cos\theta}\right) \quad (23)$$

TABLE II. COLLISION SITUATIONS WHEN Β ϵ (90°,270°)

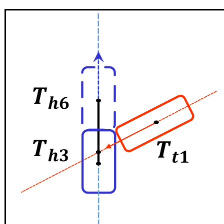
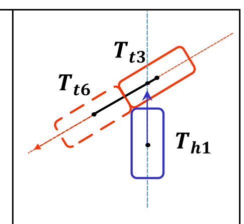
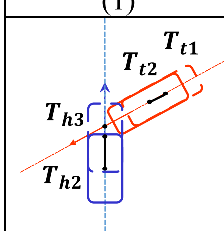
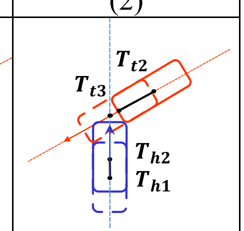
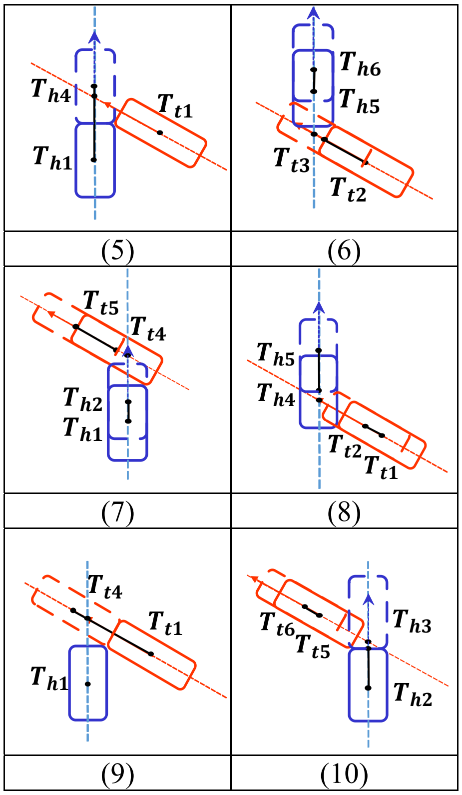
	
(1)	(2)
	
(3)	(4)

TABLE III. COLLISION SITUATIONS WHEN $\beta \in (0°, 90°) \cup (270°, 360°)$



As shown in TABLE III, angle between the two vehicles heading is less than 90 degrees. Due to the symmetry of the model, when target vehicle approaching from the left $\beta \in (270°,360°)$, TTC is calculated in the same way as $\beta \in (0°,90°)$.

IV. EXPERIMENTAL RESULT

A. *Advanced Methodology*

This section will describe the advanced topics about filtering invalid collision situations, which could be leveraged to reduce the consumption of the computation resources in order to meet the requirement of time sensitive V2X applications.

Generally, the orientation angle α of the two vehicles and the angle β between the vehicle's driving direction are calculated once GPS information is updated in host vehicle and target vehicle. Distances between two vehicles to the intersection point are also needed for TTC calculation according to our above proposed algorithm. However, there is no need to calculate the TTC all the time along the vehicle tracking when GPS location changed. Most collision avoidance TTC calculation are not necessary (also called invalid collision situation), since no actual collision will occur and some situation is not severe enough for V2X applications. For example, vehicle has not reached the intersection yet and the other vehicle has exits the intersection already. Heading and orientation filtering are introduced to eliminate these vast majority of invalid collision situations. Through which, TTC of the remaining determined collision will be calculated according to types of key timing we discussed in the section III. The invalid collision situation filtering method is mainly divided into two parts, the first part is the heading and orientation filtering, the second part is the key timing filtering.

1) *Heading and orientation filter*

In a real environment, the proportion of collisions stands a very small probability. TTC is only calculated for some special situations, in most cases, calculation could be filtered out according to the filter configurations. In order to filter out a collision-free situation, direction and coordinate filtering are used first. As shown in Fig. 3, the location of the target vehicle can be divided into four quadrants according to angle α. When the target vehicle is distributed in each quadrant, the target vehicle can be further divided into four quadrants again according to angle β between the heading of two vehicles, so all situations are divided into 16 cases.

Among which, collisions can happen within the 6 situations in non-shaded area, with 4 types of $\beta \in (0°,90°) \cup (270°,360°)$ and 2 types of $\beta \in (90°,270°)$. And 6 situations meet the following conditions respectively:

- $\alpha \in [0°,90°)$, $\beta \in [270°,360°)$
- $\alpha \in [0°,90°)$, $\beta \in [180°,270°)$
- $\alpha \in [90°,180°)$, $\beta \in [270°,360°)$
- $\alpha \in [180°,270°)$, $\beta \in [0°,90°)$
- $\alpha \in [270°,360°)$, $\beta \in [0°,90°)$
- $\alpha \in [270°,360°)$, $\beta \in [90°,180°)$

The probability of a collision occurs in the eight regions of the shaded area is minimal and only in a few cases. X_c calculated in this case will be marked as minus and the T_6 values in these regions will be calculated first. If the negative T_6 values are calculated, indicate that the current status is safe. For the 2 deep shaded regions, the probability of a collision is zero, so TTCs in these areas will not be calculated.

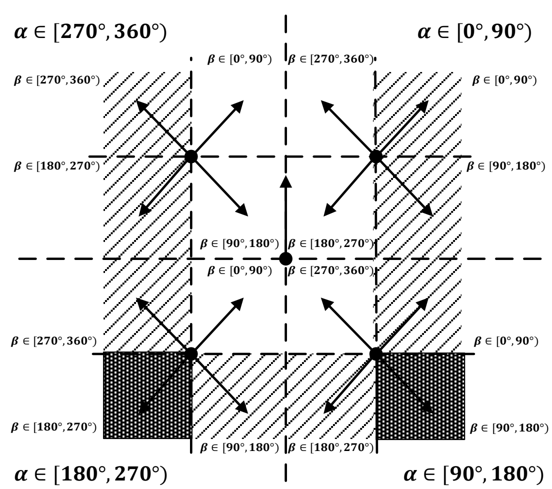


Figure 3. Heading and orientation filte

2) Key timing filter

After heading and orientation filtering in Fig. 3, we can further filter out the absence of collisions by using the key endpoints of both vehicles enters and exits the intersection. If T_{sh} and T_{st} respectively indicate the time when the vehicle and the target vehicle start to enter the intersection (refer to T_1 in Table I), let T_{eh} and T_{et} denote the time it takes for the host vehicle and target vehicle to exit the intersection (refer to T_6 in Table I). The situations when a vehicle has left the intersection and the other vehicle has not reached the intersection yet can be determined by the following formula (24):

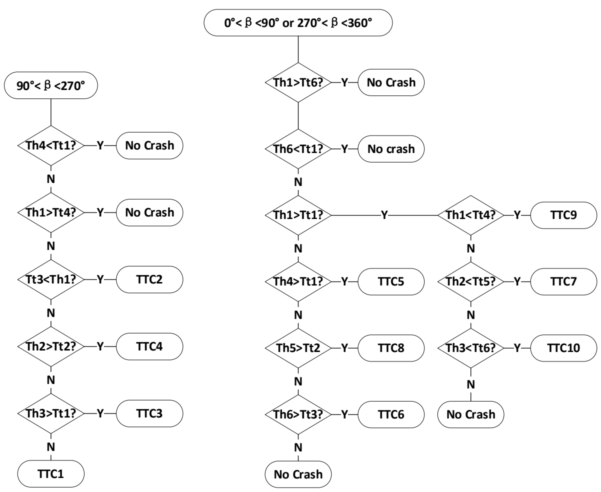


Figure 4. Collision situation discrimination process.

$$M = (T_{sh} - T_{et}) \cdot (T_{st} - T_{eh}) \qquad (24)$$

When M<0, means that one vehicle has left the intersection while the other vehicle has not yet reached the intersection, so there will be no collision. After filtering out cases of M<0, finally, according to the key time points calculated in the previous section, valid collision situation could be determined by the above process in Fig. 4.

B. Experiments

In our experiments, C/S based architecture is used to simulate the V2X data transmission. The client vehicle acquires high-precision GPS information and transmits it to the backend server through cellular network. Then the server broadcast received information through websocket to other vehicles within a certain distance around the host vehicle. Meanwhile, all the client vehicles' GPS information are kept up-to-date into the database according to timestamp for further tracking and analyzing. Two test client vehicles are experimented in the Vires [14] [15] simulation platform with visualized movement and warning information. The length and width of the two test vehicles are 4.8 meters, 5.1 meters and 1.9 meters and 2.1 meters respectively. In order to facilitate the demonstration, contradistinction is made by different time slices with different vehicle positions.

As Fig 5-7 illustrate, three experimental cases are validated by using different TTC calculation model (point model [7], circle model [10] and our proposed rectangle model) respectively. Fig.5 illustrates a normal case that two vehicles driving with different headings could trigger the collision warning through all the TTC calculation models. Rectangle model obviously gives collision situation precisely than the other models. Fig.6. shows a special case that two vehicles pass through the intersection point one after another without colliding. It can be seen that our proposed model in this paper can clearly identify this situation without issuing false alarms. Point model and circle model give the wrong collision point and provide a false collision warning, which could result into very poor user experience. Fig.7. illustrates another special case that vehicles are driving side by side in different lanes. The circle model cannot perfectly describe the true shape of the vehicle. It will also cause a lot of false alarms in practical applications. Compared with point model and circle model that frequently trigger false alarms, our proposed algorithm can be better applied to vehicle borne collision warning applications.

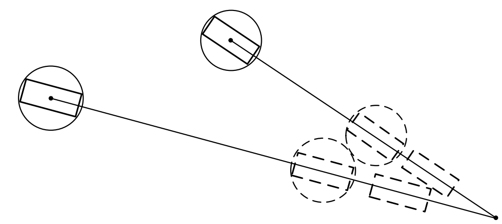


Figure 5. Normal case scenario.

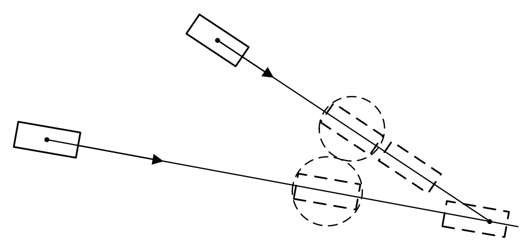


Figure 6. Special case 1 scenario.

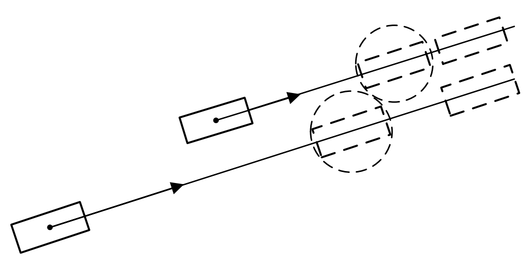


Figure 7. Special case 2 scenario.

Table IV. shows the numerical comparison results in detail. The result corresponding to case Fig.5, as an example, shows that TTC is 10.39s by using the point model in paper [7] to describe the vehicle as a point. Circle model, proposed in paper [10] describing the vehicle as a circle, gives TTC as 5.35s. In contrast, our proposed method in this paper for TTC is 7.47s. Compare to the actual situation, there is a large TTC calculation error between the collision position calculated by the point model and circle model, both models fail to achieve an accurate warning effect and fail to reduce the risks of collision. Our proposed rectangle model, on the other hand, can provide accurate TTC and avoid false alarms.

TABLE IV. EXPERIMENTAL COMPARISON BETWEEN DIFFERENT TTC

Case Number	Vehicle Status						TTC warning		
	Length(m)	*Width(m)*	*Latitude*	*Longitude*	*Heading*	*Speed(m/s)*	*Point(s)*	*Circle(s)*	*Rectangle(s)*
Normal Case Fig. 5	5.1	2.1	121.61139076	31.25956982	105.1	4.25	10.38963	5.35136	7.47499
	4.8	1.9	121.61155024	31.25961488	122.9	3.21			
Special Case 1 Fig. 6	5.1	2.1	121.61142464	31.25955344	100.8	9.72	4.03145	1.99525	None
	4.8	1.9	121.61156343	31.25962364	122.7	3.44			
Special Case 2 Fig. 7	5.1	2.1	121.61142565	31.25955221	75.9	8.27	6.86413	3.39720	None
	4.8	1.9	121.61156991	31,25962113	76.0	5.94			

TABLE V. COMPUTATION TIMES FOR NORMAL CASE TEST

Algorithm Type	Total Computation Time(ms)	Average One-shot Computation Time(ms)
Point Model [7]	35.64	2.97
Circle Model [10]	67.92	5.66
Proposed Rectangle Model (No Filter)	108.24	9.02
Proposed Rectangle Model (with IV.1 Filter)	73.44	9.18
Proposed Rectangle Model (with full filters)	48.96	6.12

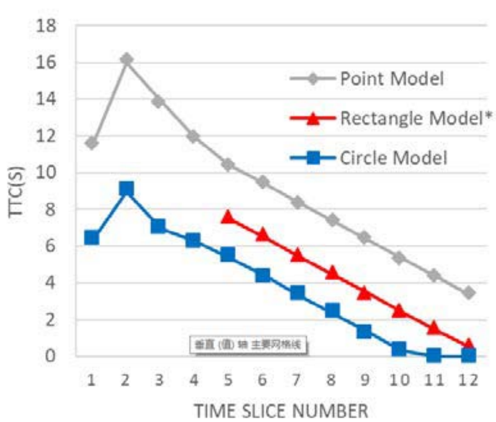


Figure 8. TTCs for normal case test.

Furthermore, we also validate the TTC calculation against different vehicle positions in different time slices as shown in Fig 8. It can be seen that the rectangle model proposed in this paper only calculates TTC when there is a collision risk. It is more efficient than other TTC calculation models. On the other hand, circle model with zero TTC result means that it wrongly predicts the vehicles being collided in earlier stage. In Table V, computation time is compared among different TTC calculation models, including our proposed rectangle model with different filters mentioned in section IV.A (no filters, heading and orientation filter, and full filters). In terms of time consumption, one-shot calculation time of the fully filtered rectangular model takes more than point model and circle model, but total time consumption is more efficient considering the precision metric. Same conclusions are validated for special cases in Fig. 6 and Fig.7.

V. CONCLUSION AND FUTURE WORK

This paper presents an accurate model for calculating TTC that models vehicles approximately as a rectangular in cross-collision situations. Compared to the point model [7], the improved rectangular model can calculate the TTC more accurately, and can also have superior performance when the angle between two vehicles is small. Rectangular model is also effective in reducing the calculation of non-collision situation,(two vehicles pass by the intersection without collision), solved the ubiquitous problem caused in the point model which eliminate the number of false alarms. Compared to the circular model and the elliptical model [9], the rectangular model has a more accurate contour that better approximates the

shape of the vehicle so as to avoid false alarms when the vehicle is traveling side by side in close proximity. For the existing rectangular model [13], the algorithm used in this paper does not use simulation method, which can effectively reduce the computational cost when the number of vehicles in the range increases. The method used in this paper also has an advantage in terms of the accuracy of the TTC since the TTC calculated by the simulated method is suitable for the time sensitive V2X applications. On the other hand, since this paper calculates the orientation angle α, it is easier to warn the user of the direction from which the threat came from. However, when the vehicle travels along a curved trajectory, the algorithm will calculate the collision in the tangential direction of the curve according to the position where the GPS is updated, which cannot completely consider the whole curve trajectory. This problem might be solved using Kalman filtering in our future work.

REFERENCES

[1] Organization W H. Global status report on road safety - 2013: supporting a decade of action.[J]. World Health Organization, 2013, 15(4):286-286.

[2] Seo, Hanbyul, et al. "LTE evolution for vehicle-to-everything services." IEEE Communications Magazine 54.6(2016):22-28.

[3] Minderhoud, M. M., and P. H. Bovy. "Extended time-to-collision measures for road traffic safety assessment." Accident; analysis and prevention 33.1 (2001):89-97.

[4] Hayward J C. Near-miss determination through use of a scale of danger [J]. Highway Research Record, 1972.

[5] Xiang, Xuehai, W. Qin, and B. Xiang. "Research on a DSRC-Based Rear-End Collision Warning Model." IEEE Transactions on Intelligent Transportation Systems 15.3(2014):1054-1065.

[6] Zhang, Ruifeng, et al. "A method for connected vehicle trajectory prediction and collision warning algorithm based on V2V communication." International Journal of Crashworthiness 22.1 (2017): 15-25.

[7] Li, Yongfu, L. Zhang, and Y. Song. "A vehicular collision warning algorithm based on the time-to-collision estimation under connected environment." International Conference on Control, Automation, Robotics and Vision IEEE, 2017.

[8] Huang, Chung Ming, and S. Y. Lin. "An Advanced Vehicle Collision Warning Algorithm over the DSRC Communication Environment: An Advanced Vehicle Collision Warning Algorithm." Cambridge Journal of Education 43.2(2013):696-702.

[9] Jia, Hou, G. F. List, and X. Guo. "New Algorithms for Computing the Time-to-Collision in Freeway Traffic Simulation Models." Computational Intelligence & Neuroscience 2014(2014):761047.

[10] Wang, Yunpeng, et al. "Vehicle collision warning system and collision detection algorithm based on vehicle infrastructure integration." Advanced Forum on Transportation of China IET, 2012: 216-220.

[11] Felipe Jiménez, José Eugenio Naranjo, and Fernando García. "An Improved Method to Calculate the Time-to-Collision of Two Vehicles." International Journal of Intelligent Transportation Systems Research11.1(2013):34-42.

[12] Huang, Chung Ming, and S. Y. Lin. "Cooperative vehicle collision warning system using the vector-based approach with dedicated short range communication data transmission." Iet Intelligent Transport Systems8.8(2014):124-134.

[13] Zhang, H., and Z. C. Li. "Forward modeling of carbonate fracture reservoir based on time-space dual variable grid algorithm." Journal of China University of Petroleum 35.3(2011):51-57.

[14] Kai F, Balaghiasef R, Düring M, et al. A Cooperative Driver Assistance System: Decentralization Process and Test Framework[J]. Annals of Noninvasive Electrocardiology the Official Journal of the International Society for Holter & Noninvasive Electrocardiology Inc, 2015, 16(2):2432-2438.

[15] Berger C, Block D, Heeren S, et al. Simulations on Consumer Tests: A Systematic Evaluation Approach in an Industrial Case Study[J]. IEEE Intelligent Transportation Systems Magazine, 2014, 7(4):24-36.