2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Automated Toll Gate Passing

Zhonglin Xu[1], Xinhui Di[1], Houming Wang[1], Jun Xu[1], Rolf Adomat[2], and Chen Zhang[1]

Abstract -- **This paper proposes a new approach of automated toll gate passing of an automated driving vehicle. This approach enables the vehicle to select an optimal toll gate and to automatically pass the toll gate by using object detection, 3D environment construction, virtual line generation, path planning and motion control. After designing the concept of the approach, some demonstrations are conducted to prove it. This data-based scenario shows that the proposed approach can not only perceive the environment well for this purpose but can also plan appropriate trajectories when encountering complex scene near toll plazas.**

I. INTRODUCTION

With the rapid development of Level 3 and Level 4 automation, lots of automated driving vehicles equipped with newest technologies appear all over the world. In line with the current trend in the automobile industry, many companies and R&D centers have already realized the automated driving in structural road of the highway.

Not only the classical OEMs and the Tiers (first group) but also the data and artificial intelligence driven companies (second group) begin to face this technological challenge. These two groups of players build a variety of automated driving systems and embed them into vehicles. The first group of players have their origin and plenty of experience in manufacturing and controlling the vehicles. They developed Advanced Driving Assistance Systems (ADAS), e.g. Adaptive Cruise Control (ACC) [1] and Lane Keeping Systems (LKS) [2], and integrated these functions step by step into the existing vehicle systems. The second group have their natural superiorities in data driven methodologies. Their automated driving systems are characterized by deep-learning based methods, both for perception and for driving strategy enabled by availability of tons of data.

The existing ADAS and automated driving systems developed by these two groups of companies ensure the secure driving on highways in normal cases. However, many complex situations without landmarks but with unpredictable actions of other road participants, such as toll gate passing, are not covered by the existing systems. Additionally, the infrastructure of and driver behavior in such toll gate plazas are different from country to country, even in Asia, e.g. China and Japan.

This paper proposes a new approach of an automated driving vehicle for automatically passing an Electronic Toll Collection (ETC) gate on highways. Information in consist of sensor, HD map, V2X is useful for redundancy, which is needed for maintaining high robustness. In our study, for creating the robust 3D environment in the tollgate area, the effectiveness by only sensor based approach is verified in this paper. It is a vision-based system in which the vehicle can perceive other road users, like vehicles and pedestrians, as well as infrastructure elements, like gates, barriers and traffic signs. A trajectory-based model is then used to generate virtual lanes because of non-existing lane markers in that area and to select the optimal lane by modeling and understanding the actual scene. It is also a traffic reactive system where the ego vehicle is able to make decision according to the dynamic traffic situation, like building of vehicle queues and cut in of other vehicles.

II. FUNCTIONAL ARCHITECTURE

From functional point of view the proposed approach consists of three parts, which are perception, decision making, and motion control, as illustrated in Figure 1.

When approaching a toll gate plaza, the first step is to perceive the environment. Both infrastructure, e.g. gates and traffic signs, and moving objects, e.g. vehicles and pedestrians, are detected by the dedicated detection function modules for diverse object classes. The detected vehicles are further categorized into vehicle queues. Afterwards, the simultaneous localization and mapping (SLAM) algorithm estimates the motion state of the ego vehicle and generates a 3D sparse map with the localization of the ego vehicle into the environment. The combination of object detection and SLAM constructs an environment in which drivable areas are derived.

In the next part, i.e. the decision making phase, traffic signs are further understood and all toll gates are scanned in order to detect passable ETC toll gates out of them which are marked as so called candidate gates. Only for these candidate virtual lanes are generated from the

[1] Zhonglin Xu, Xinhui Di, Houming Wang, Jun Xu, and Chen Zhang are with the Continental China (email: Zhonglin.Xu, Xinhui.Di, Houming.Wang, Eric.Xu, Chen.2.Zhang@continental-corporation.com).
[2] Rolf Adomat is with the Continental Japan (email: Rolf.Adomat@continental-corporation.com).

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

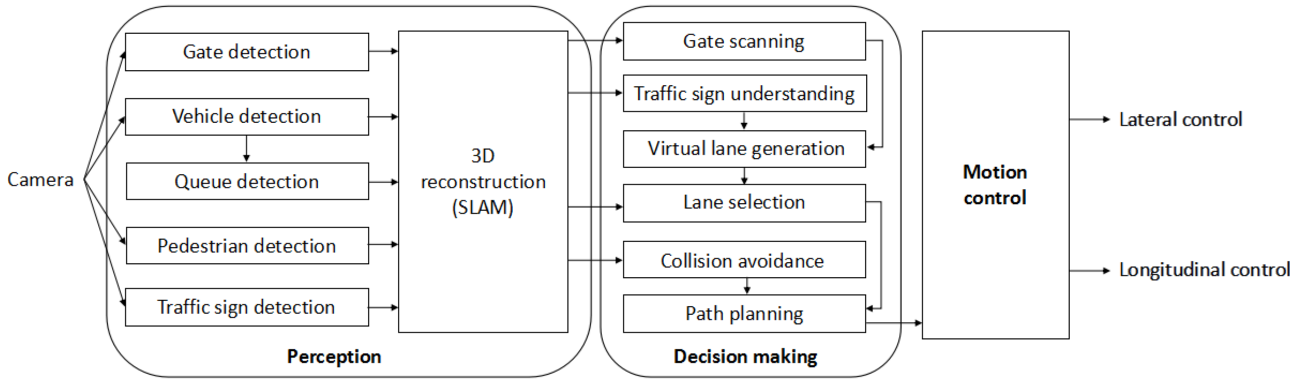

Figure 1. Functional and logical architecture with three parts of sub-functions and the functional module in them: perception, decision making and motion control.

current location of the ego vehicle as starting point and with the gate entry points of candidate gates as the ending points, with the same number of virtual lanes as candidate gates. From all the virtual lanes, the lane with the shortest vehicle queue and minimum lateral distance in total is then selected as the optimal virtual lane. Finally, a collision-free trajectory by considering drivable areas is permanently calculated and transmitted to the motion control module in order to maneuver the automated driving vehicle to the target toll gate.

III. PERCEPTION

This section presents the functional modules related to perception, which flowchart is shown in Figure 2. An object detection algorithm based on convolutional neural network (CNN) is used to sense other vehicles, pedestrians and ETC gates. The correlation between the vehicles is calculated to determine vehicle queues. Finally, the SLAM is used to construct a 3D environment model by incorporating the detected objects as well as vehicle queues.

A. Detection of Road Users and Infrastructure Elements

Convolutional neural networks have shown huge success in object recognition. From 2014 to 2016, lots of CNN-based algorithms including R-CNN [3], Fast R-CNN [4], Faster R-CNN [5] and YOLO, achieve more and more accurate and high-speed results. Many deep neural network structures, such as GoogleNet [6], have been adopted. In the proposed approach, YOLOv2 [7] is implemented in terms of a single convolutional neural network for detecting vehicles and pedestrians as well as ETC gates because of its accuracy and real-time property.

The structure of YOLO algorithm, referring to the structure of GoogleNet, is divided into 24 convolutional layers and 2 connected layers, which uses 1×1 reduction layer followed by 3×3 convolutional layers to substitute the inception modules in GoogleNet. YOLO algorithm predicts positions and class probabilities directly from full images. The input image is firstly segmented into $S \times S$ cells, each of which predicts k bounding boxes with confidence as well as C conditional class probabilities. Furthermore, each bounding box is represented by a coordinate (x, y, w, h, confidence). The (x, y) are the center offset between the bounding box and the bounds of the grid cell. The w is the width and h is the height respectively. The confidence is defined by Pr(object)×IOU. The Pr(object) is 1 when a grid cell contains a part of a ground truth box, vice versa. The IOU is the intersection between the predicted bounding box and the ground truth box. By these predictions, the class-specific confidence score of each bounding box can be obtained and finally the bounding boxes having high scores in each grid cell to predict objects in the image are selected globally.

The YOLO model is trained by ImageNet [8] dataset, which is able to detect over 9000 categories of objects including vehicles and pedestrians. The weight of the model is fully used and continuously trained by the ETC dataset and manual gate dataset collected from Shanghai City to Hangzhou City, the total amount of which is 500 and 300 respectively. After the training, the model can detect the ETC signs, manual signs, the vehicles and pedestrians.

B. Queue Modeling

This functional module is used to estimate the length of the queue at each candidate gate by calculating the correlation between the neighboring vehicles. The lengths of the queues are part of the decision basis for selecting the optimal candidate gate.

In this module, the progressive correlation processing is proposed as a means to realize an efficient method to estimate the vehicle queue. It is robust to several situations including heavy and light shadows. Results

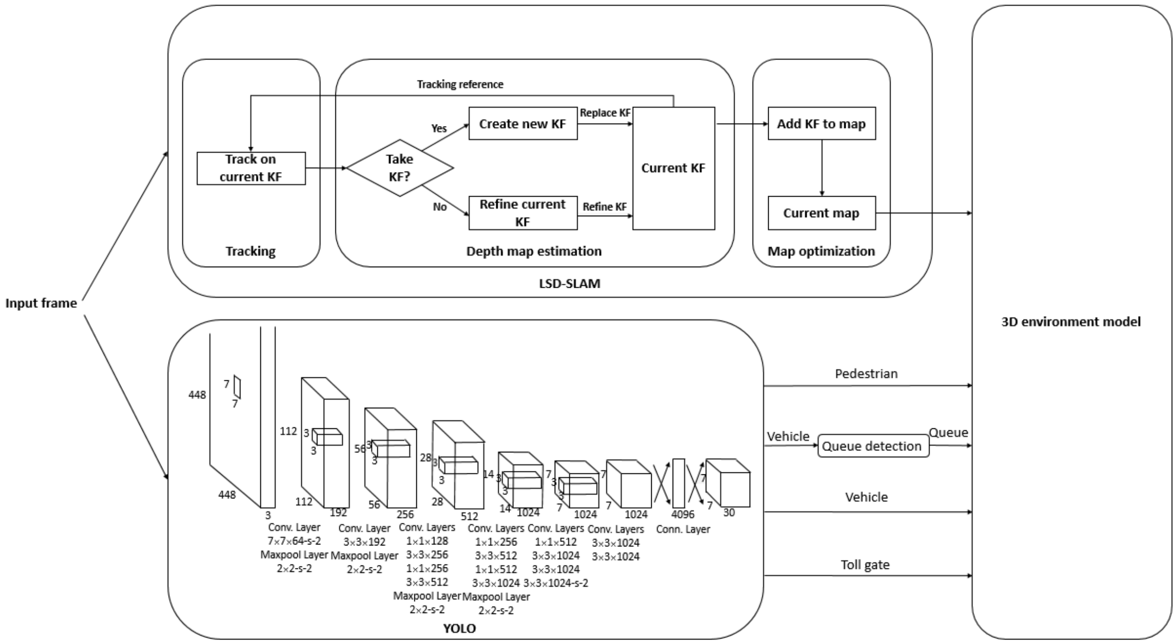

Figure 2. The flowchart of the generation of 3D environment model. The input image is processed in two paths. The first path is the LSD-SLAM part, which generates the 3D sparse map with key features (KF). The second path is the YOLO part, which detects the infrastructure elements and the road users. Results of both paths are fused into the 3D environment model.

show that this method has advantages in accuracy as well as robustness.

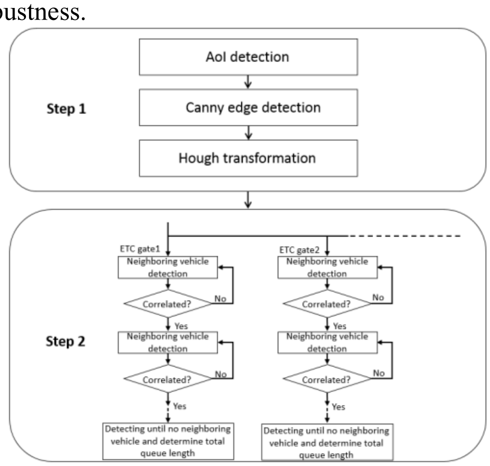

Figure 3. The flowchart of progressive correlation algorithm.

This method is divided into two steps, which flowchart is shown in Figure 3. The first step is to detect the area of interest (AoI), which restricts the road surface and discards the useless regions like sky and buildings. The Canny edge method is to detect the edges in the image. Based on the edges, Hough transformation determines the straight line, which can be the road border or potential landmarks. The second step is to determine the total queue length to each candidate gate. For each candidate gate, the algorithm firstly detects the neighboring vehicle 1 and calculate the correlation of the vehicle1 and the candidate gate. The correlation calculation is determined by the position correlation and overlapping area. If correlated, the neighboring vehicle2 is detected and the correlation of the vehicle1 and vehicle 2 is calculated. This process is done step by step to finally determine the total queue length. This method has the ability to calculate the total queue length even if some vehicles in the queue are occluded by the closer vehicles.

C. 3D Environment Modeling

SLAM is used to construct or update a map of an unknown environment, while simultaneously keeping track of an ego vehicle's location in the map. In the proposed approach, the Large-Scale Direct Monocular SLAM (LSD-SLAM) [9] is applied. It directly operates on image intensities both for tracking and mapping. The flowchart is shown in upper part of Figure 2, which can be divided into 3 steps: tracking, depth map estimation and map optimization.

The combination of positions of detected objects and depth information from SLAM algorithm makes up to the 3D environment model. In order to secure safety, the predicted path of detected vehicles is also calculated in this model.

IV. DECISION MAKING

In this step, the decision in terms of selection of an optimal virtual lane out of all possible ones is made. In the first step, all ETC gates and general gates are filtered out and traffic signs are understood for being able to obey the traffic rules when approaching these candidate gates. A virtual lane is defined as the optimal virtual lane if the length of its vehicle queue and the lateral distance to that vehicle queue from the ego vehicle in total is minimal. Having decided for a virtual lane as the optimal one, the vehicle is controlled to pursue this lane. In doing so, dynamic changes in the surrounding, e.g. other approaching vehicles into the queues and possible cut-ins of other vehicles, are also considered. The output of this step is the best trajectory the vehicle should go within the optimal virtual lane.

A. Gate Scanning

After the 3D environment is established, gate scanning is used to detect all the toll gate and filter out all the passable ETC gates while discarding unpassable manual gate. Because of the possibility of toll gate occluded by trucks or buses, it could take a while to get all the gates scanned and all the ETC gates determined.

B. Virtual lane Generation

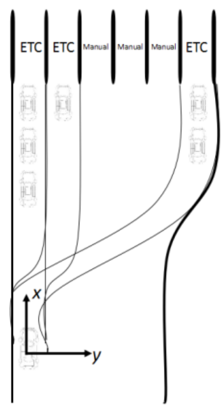

Figure 4. This figures illustrates generation of virtual lanes in a toll gates plaza. Only to ETC gates virtual lanes are generated. The position of ego vehicle is the origin of a moving Cartesian coordinate system.

This module generates the virtual lanes from the position of ego vehicle to all determined ETC gates, first time carried out when arriving at the toll gate plaza without lane markings.

Hereby the road is transformed into a Cartesian coordinate system. Each point on the road has a coordinate. The position of ego vehicle is the coordinate origin, as shown in Figure 4. If the number of ETC gates is N. The coordinates of the ETC gates are: (x_1, y_1), (x_1, y_2), …, (x_1, y_N).

For each ETC gate, the cubic function family

$$y= ax^3+bx^2+cx+d$$

is used to interpolate the y coordinate between the coordinate origin and the ETC gates. The best cubic function is chosen according to the driving distance and the size of available spaces.

C. Lane Selection

This module is used to select the optimal virtual lane from several virtual lanes to the candidate gates.

In order to find the best solution, two factors need to be considered, the lateral distance to each candidate gate and the length of the vehicle queue of each candidate gate. Particle filter [10] is used to estimate the probability of each factor for the reason that it is based on Monte-Carlo methods, which can be used in any form of state space model. Large probability means a high chance to choose corresponding candidate gate.

For 1, 2, 3, ..., k denote each candidate gate. The probability vector $\mathbf{P}_{\mathrm{N}} = (p_n(1), p_n(2),..., p_n(k))$ and $\mathbf{P}_{\mathrm{L}} = (p_l(1), p_l(2),..., p_l(k))$ denote the particle filter result of lateral distance and the particle filter result of vehicle queue length respectively. The final decision probability vector $\mathbf{P}$ is combined of $\mathbf{P}_{\mathrm{N}}$ and $\mathbf{P}_{\mathrm{L}}$ by Yager formula [11], which is an information fusion algorithm.

In our approach, the maximum probability in $\mathbf{P}$ is chosen to select the final ETC gate.

D. Collision Avoidance

This functional module is used to permanently monitor road users in the surroundings if they could bring danger to the ego vehicle. The determined free space exclusive area of dangers is used for path planning in the next step.

E. Path Planning

This module is used to step by step plan a collision-free trajectory for the motion control.

The optimal virtual lane is transformed into *S-L* coordinate system, shown in Figure 5. *S* is the direction of the path, while *L* is the vertical direction of *S*. The whole virtual line is segmented into a series of cells. Each cell has a cost. The cell with small value means it is suitable for driving, while the cell with large value means it is close to the edge of the virtual lane or is closed to the vehicles and pedestrians, which is not appropriate for driving.

Let n denotes the cell. k denotes the number of the ETC gates. Considering n_0, n_1, …, n_k consist a path τ. n_0 is the origin point and n_{k-1} is the current point. n_k is the cell to be chosen. The cost from n_0 to n_k is:

$$\Omega(\tau(n_0, n_1, \ldots, n_k))=$$
$$\Omega(\tau(n_0, n_1, \ldots, n_{k-1}))+ c(\tau(n_{k-1}, n_k)) +\Phi(n_k)$$

$c(*)$ is the cumulative cost along the trajectory from n_{k-1} to n_k. $\Phi(*)$ represents the cost of n_k.

The n_k chosen is defined by:

$$n_k \leftarrow \underset{n_k}{\arg\min}\, \Omega(\tau(n_0, n_1, \ldots, n_k))$$

n_k is chosen step by step, which can avoid collision or potential danger.

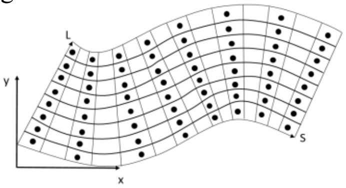

Figure 5. The *S-L* coordinate system in path planning. The black dots are the cell centers.

V. MOTION CONTROL

Motion control is used to manage the vehicle's longitudinal and lateral control which adopts an adaptive method. It receives a collision-free trajectory data from path planning such as curvature, yaw rate and velocity. In the proposed approach, LQR-PID [12] algorithm is used.

The full vehicle model is simplified into a bicycle model. Then a state feedback controller is brought into our dynamic bicycle model, so as to configure a new state transformation matrix. With complex regulation of the feedback matrix, it comes to the pole of the closed loop system. However, LQR can exactly pave a way to the optimum pole. The whole part of LQR predicts an expectation as inputs to PID controller. And PID corrects the system response according to the deviation which results from comparisons between expectation and actual value. LQR controller can predict an optimum expectancy for PID with data from the CAN bus. PID is a classic control with strong adaption and robustness. This combination of optimum control and stable control shapes the performance of the proposed motion control, which is more secure and robust than PID.

VI. RESULTS

In order to prove the feasibility of the described approach of the automated toll gate passing, a real-world test was conducted. In this test, perception, decision making and motion control parts were working as described, however in an open loop system, meaning the result of motion control was not linked to the actuators. Instead, the driving was performed by a human driver. The test zone was the toll gate plaza Fengjing, located on the highway from Shanghai to Hangzhou, with 2 manual gates and 3 ETC gates. During the test, the ego vehicle speed limit was at a maximum of 25 kph in toll gate plaza.

A. Object Detection

The YOLO algorithm is used here to detect objects. The already trained weights of YOLOv2 are used to detect vehicles and pedestrians. In order to detect ETC gates, the deep neural network is further trained by 500 ETC gate datasets and 300 manual gate datasets, collected from Shanghai to Hangzhou.

A video stream during passing of the toll gate plaza Fengjing is used for testing. The size of each frame of the video is transformed into 640×480 pixels, which is suitable for the YOLO algorithm to deal with. Figure 6 shows the results of object detection, in which can be seen that the YOLO algorithm can detect most of the vehicles. When the ego vehicle is close enough to the ETC gate, the algorithm is able to detect ETC signs.

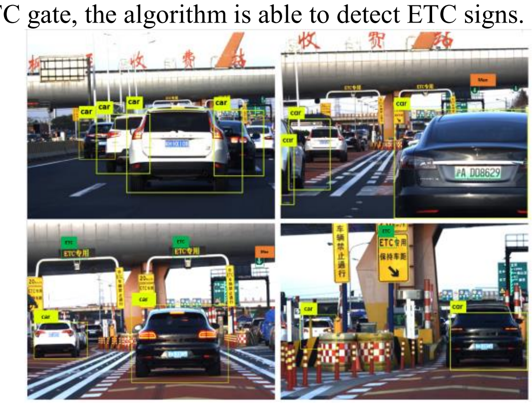

Figure 6. Object detection results. The detected cars are marked by yellow rectangles. The detected ETC signs are marked by green rectangles.

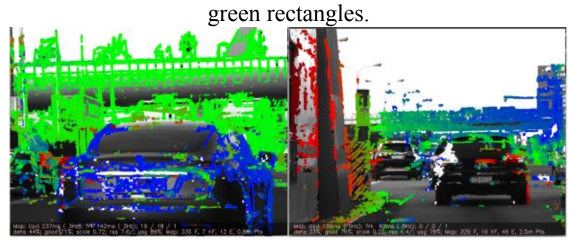

Figure 7. The 3D sparse maps of LSD-SLAM. The depth information is shown in different colors.

B. SLAM Algorithm

The same video stream of object detection is used to demonstrate the LSD-SLAM algorithm. The size of each frames of the video is also transformed into 640×480 pixels. In order to accelerate the speed of the algorithm, color frames are converted into gray images. The demonstration of the results from the LSD-SLAM algorithm is shown in Figure 7, which are 3D sparse maps. Different color means different depth. The Results show that the LSD-SLAM algorithm can properly generate a 3D sparse map.

C. Decision Making

Finally, decision making is demonstrated. Figure 8 is the scheme of the real scene during the toll gate passing.

The automated driving vehicle approaches the toll gate. Five toll gates are detected. Two ETC gates locate on the left side, while another one locates on the right side. In parall, the ego vehicle detects other vehicles in the surroundings. For each ETC toll gate, a virtual lane is generated, shown in blue line in Figure 8(a). Because of the shortest vehicle queue and the least lane changing in the second ETC toll gate, the ego vehicle selects the second virtual lane from left, shown in Figure 8(b). Considering to the situation of the second virtual lane, the ego vehicle plans step by step a path. When another vehicle, indicated as no. 2, rushes into the virtual line, the ego vehicles can take actions such as braking to avoid collisions, shown in Figure 8(c). Finally, the ego vehicle passes the toll gate along the virtual lane.

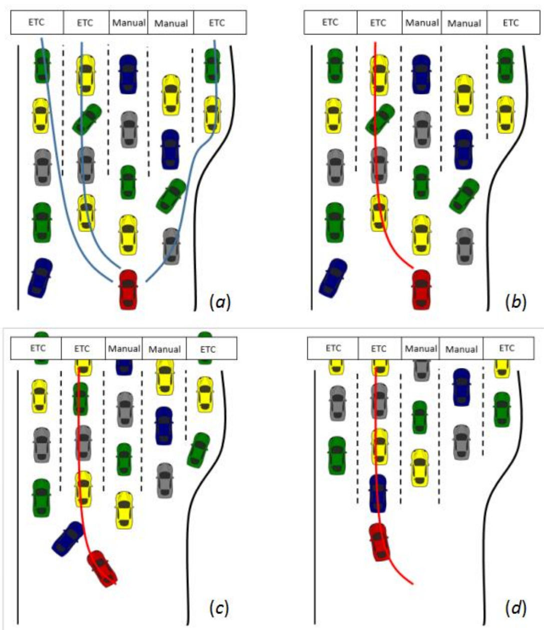

Figure 8. Decision making scheme. (*a*) Virtual lanes generation. (*b*) Virtual lane selection. (*c*) The reaction of ego vehicle to happenings. (*d*) Toll gate passing.

VII. CONCLUSION AND FUTURE WORK

This paper proposes a new toll gate passing approach, which is divided into perception, decision making and motion control. In the perception part, the deep neural network is used to detect the toll gates, vehicles and pedestrians. In parallel, the depth of the detected objects are obtained to establish a 3D environment model. In the decision making part, virtual lanes are generated and the optimal lane is selected. A collision-free path is planned and transmitted to the motion control part to maneuver the vehicle. The effectiveness of object detection and SLAM algorithm is shown in this paper, and the ability of automated driving vehicle to plan an appropriate path and to pass the toll gate.

In the future, further input data, e.g. V2X signals from infrastructure or other road users and a HD map of the area, can be used to reduce the complexity of the situation and to increase the capability of predictive planning. Getting directly a signal from an ETC gate via V2X will make the visual classification of an ETC gate obsolete. Having a HD map of the toll gate plaza, virtual lanes can be planned in the very early phase. In doing so, many critical situations can be prevented to happen. However, V2X and HD map can only reduce the complexity but not solve the core of the problem, because even HD maps will not contain the dynamic building of vehicle queues, and it is unforeseeable when all road users will be equipped with V2X technology. Thus, the proposed system is designed to be more universally usable, even without HD map and V2X.

VIII. REFERENCE

[1] N. Liberis, C. Roncoli, M. Papageorgiou, “Predicotor-Based Adaptive Cruise Control Design”, IEEE Transactions on Intelligent Transportation Systems, 2017.

[2] C. Kang, J. Lee, S. Yi, S. Jeon, Y. Son W. Kim, S. Lee, “Lateral Control for Autonomous Lane Keeping System on Highways”, 15th International Conference on Control, Automation and Systems, 2015.

[3] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[4] R. Girshick, “Fast R-CNN”, Proceedings of the IEEE International Conference on Computer Vision, 2015.

[5] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp 1137-1149, 2017.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “Going deeper with convolutions”, IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[7] J. Redmon, A. Farhadi, “YOLO9000: Better, Faster, Stronger,” IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, L. Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 2015.

[9] J. Engel, T. Schops, D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular Slam”, Springer International Publishing, pp 834-849, 2014.

[10] P. Moral, “Non-Linear Filtering: Interacting Particle Solution,” Markov Processes & Related Fields, pp 555-580, 1996.

[11] Yager R. “On the Dempster-Shafer Framework and New Combination Rules,” Information Science, 41 (2):93-137, 1989.

[12] L. Argentim, W. Rezende, P. Santos, R. Aguiar, “PID, LQR and LQR-PID on a Quadcopter Platform”, International Conference on Informatics, Electronics and Vision, 2013.