The 4th Annual IEEE International Conference on
Cyber Technology in Automation, Control and Intelligent Systems
June 4-7, 2014, Hong Kong, China

# Local path planning in a complex environment for self-driving car

Unghui Lee
Dept. of Aerospace Engineering
KAIST
Daejeon, Korea
lamer0712@kaist.ac.kr

Sangyol Yoon
Dept. of Aerospace Engineering
KAIST
Daejeon, Korea
sangyol.yoon@kaist.ac.kr

HyunChul Shim
Dept. of Aerospace Engineering
KAIST
Daejeon, Korea
hcshim@kaist.ac.kr

Pascal Vasseur
LITIS
Université de Rouen
France
pascal.vasseur@univ-rouen.fr

Cédric Demonceaux
Le2i
Université de Bourgogne
France
cedric.demonceaux@u-bourgogne.fr

***Abstract*—This paper introduces an local path planning algorithm for the self-driving car in a complex environment. The proposed algorithm is composed of three parts: the novel path representation, the collision detection and the path modification using a voronoi cell. The novel path representation provides convenience for checking the collision and modifying the path and continuous control input for steering wheel rather than way point navigation. The proposed algorithm were applied to the self-driving car, EureCar(KAIST) and its applicability and feasibility of real time use were validated.**

***Keywords—Self-Driving; Autonomous vehicle; Path Planning; Obstacle Avoidance; Voronoi Cell***

## I. INTRODUCTION

A driver assistance system is developed to assist driver’s safety such as lane keeping [1], automatic parking [2], and front vehicle collision prevention [3]. This system has been applied to many vehicles and helped reduce traffic accident rate. Furthermore, recently a number of auto motor companies have developed a self-driving car and active R&D is undergoing to achieve commercialization in the near future.

One of the reasons for rapid advancement of a car that is capable of autonomous driving in a real road while keeping all traffic signals is due to the Grand Challenge and Urban Challenge organized by the DARPA (Defense Advanced Research Projects Agency) [4][5]. The DARPA established the Grand Challenge for the first time in 2004 where unmanned vehicles run the desert followed by the second Grand Challenge in 2005 and the Urban Challenge in 2007. The series of competitions have been a great success, providing impetus for unmanned vehicle technology advancement and research and development around research centers and universities over the USA as well as around the world.

A self-driving car performs path planning and obstacle avoidance utilizing the surrounding environment information obtained via radar, laser, and image sensors that can detect obstacles and drives to the goal without intervention of a driver [6].

Path planning determines vehicle's behavior and can be divided into global path planning that drives car according to the preset path information and current position of a car and local path planning that responds to signals from sensors regarding the surrounding environment.

There have been a number of studies regarding path planning, which can be categorized into Potential-field approach [7], Roadmap based approach [8][9][10][11], and Cell decomposition based approach [12].

The cell decomposition based approach is to divide free space where no collision against obstacles is found into a certain size of cells and find a path by connecting adjacent cells. Recently, a study on path planning using this approach that can be applicable to U-turn and parking in consideration of nonholonomic constraints has been published [13]. However, the result of path planning using the cell decomposition based approach is dependent on cell’s type and size too much and it is to be not suitable for path planning for high speed driving.

The roadmap based approach is divided into deterministic roadmap [8][9]and probabilistic roadmap [10][11]. A typical example of the probabilistic roadmap is using RRT. In this approach, an algorithm that considers nonholonomic motion of vehicles is widely used [11]. However, in order to process the RRT algorithm in real time, high computing is required and its result is highly dependent on heuristics. Above all, it is difficult to predict the result of path planning thereby causing insecurity, which is the main drawback of this approach.

In this paper, a new path representation method and a method of collision detection with path representation are proposed. A path planning algorithm using Voronoi Cells [9] as the deterministic roadmap approach was developed. Through simulations and experiments, its applicability and feasibility of real time use were validated.

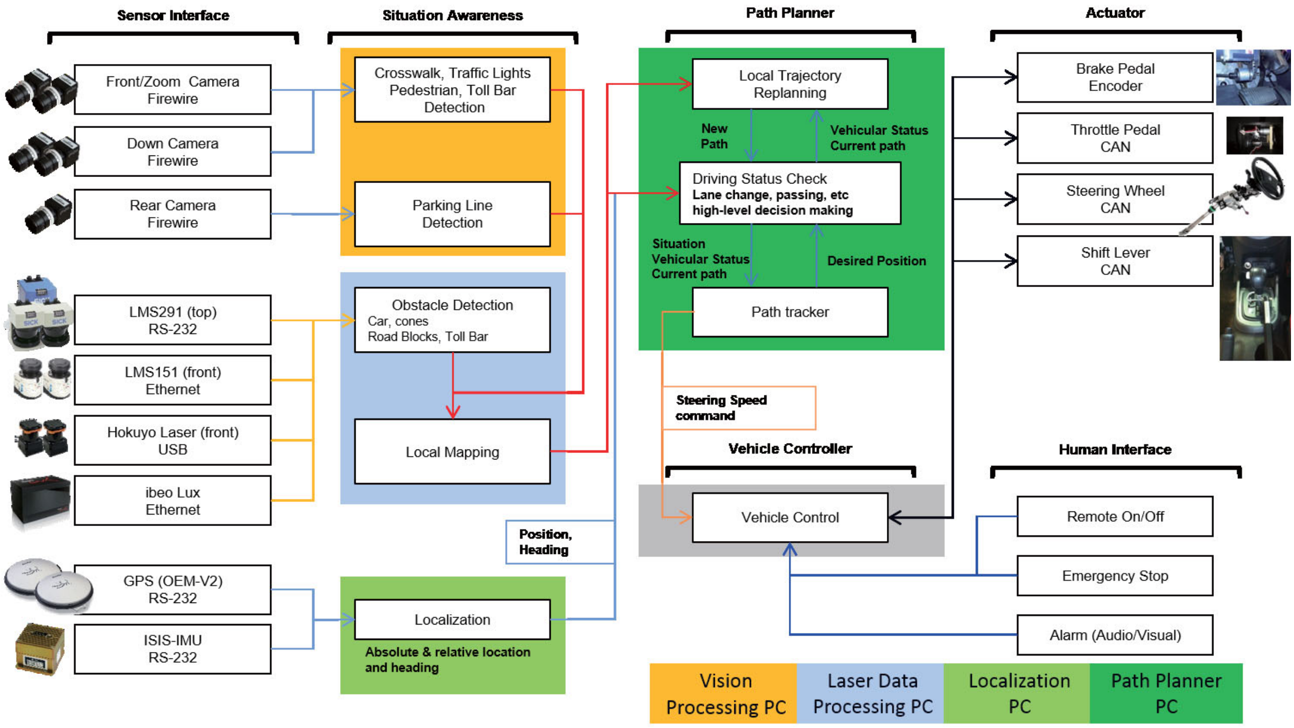


Fig. 1. KAIST Self Driving Car, EureCar, System Configuration

SYSTEM OVERVIEW

For the experiment of path tracking and generation, a vehicle developed by KAIST EureCar team (Figure 2) was used. The vehicle is equipped with two DGPSs from Novatel Company, IMU (Inertial Measurement Unit) and industrial PC104 for navigation, eight laser scanners for collision detection and five cameras for image processing. Three PCs equipped with i7 CPU were also used for laser data and image signal processing, path management and generation, and control input computation. To manipulate the vehicle, an actuator was mounted to throttle and brakes while the handling used MDPS controlled by CompactRio from NI. The system configuration is shown in Figure 1.


Fig. 2. KAIST Self-Driving Car, EureCar

## II. PATH REPRESENTATION

A way point-based navigation [14], which is a navigation algorithm for path tracking, has been widely used for robots that drive slowly or require simple direction change. In the case of a vehicle traveling at high speeds, unstable motion can be caused due to discontinuous control input occurred by change of way points. Way points are inappropriate representations for continuous collision checking along the path and modification for avoidance. Thus, this paper defined continuous path using polynomials and presented vehicle control using existing way point tracking algorithm.

### A. Path representation

A path is configured as n cubic equations once given way points are divided into certain sections. In order to prevent coefficients of the cubic equations from defining infinitely, a starting point of each path sections is positioned to an origin and a coordinate is transformed to make a slope to become zero, thereby performing curve fitting into cubic equations (1). Under the above conditions, low order coefficients of the cubic equations become zero

$$y_i = a_i x_{path}^3 + b_i x_{path}^2 \tag{1}$$

$$P_i = [x_{i,ENU}, y_{i,ENU}, h_{i,ENU}, a_i, b_i, L_i, V_i]^T \tag{2}$$

A i-th path can be expressed simply by an array in (2), which reduces the size of the array significantly compared to defining it by way points and can be used for search of existence of collision or new path generation conveniently. $x_{i,ENU}$, $y_{i,ENU}$ and $h_{i,ENU}$ represent starting position and azimuth of path while $L_i$ and $V_i$ represent a length of path and maximum speed (limit speed) of corresponding path.

### B. *Path tracking*

The present study assumed no slip between wheels and surface, and used a model which assumes the right and left wheels as one to control lateral direction of a vehicle. Equation (3) shows the induction of the rotation radius of a vehicle into the central axis of the rear wheel in a vehicle using kinematics. $L$ and $\delta_f$ represent wheelbase and steering angle.

$$R = L / \tan(\delta_f) \tag{1}$$

$$R = -0.5[(x_{con,body})^2+(y_{con,body})^2] / y_{con,body} \tag{2}$$

To track the path, one point over the path is selected as a control point. The control point is designated to a position where $D(V)$ is variably changed according to the vehicle speed in the front side of the driving direction (Figure 3). A control point $X_{con}$ over the path can be transformed into a body coordinate system via coordinate transformation. Equation (4) shows a relationship between rotation radius of a vehicle and control point in the body coordinate system. A steering angle, $\delta_f$, which makes a vehicle to reach to the control point, can be obtained through (3) and (4).

## III. PATH PLANNING

Path planning is divided into collision detection over the path and path generation for obstacle avoidance. Collision detection is determined by processing obstacle data obtained from the scanner and judging whether obstacles is crossed over the path. Path generation creates a new alternative path by modifying the existing path to avoid obstacles that block the path.

### A. *Collision detection*

Collision detection algorithm is divided into two parts: detection from vehicle position to the control point in Figure 3, and detection from the control point to the sensing area.

Collision detection from a vehicle to the control point is conducted for preparing the speed profile for vehicle's deceleration and emergency stop. This detection determines existence of obstacles over the trajectory generated according to the rotation radius of a vehicle between current vehicle position and the control point.

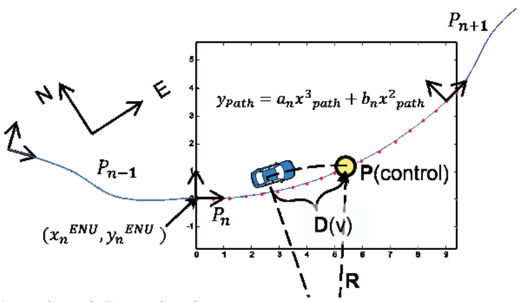


Fig. 3. Path and Control Point

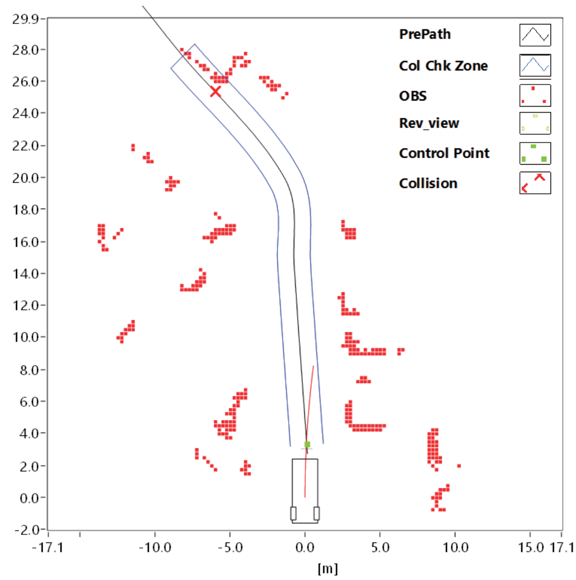


Fig. 4. Collision detection

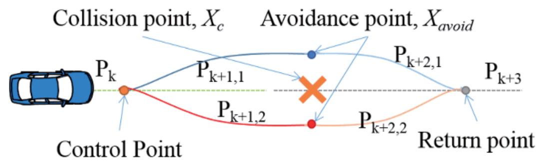


Fig. 5. Collision avoidance

The collision detection algorithm from the control point to the sensing area determines existence of obstacles by extending the path to the right and left side to create a collision detection boundary based on the assumption that vehicle follows the path correctly as shown in Figure 4.

The right and left boundary can also be expressed by path representation. The collision can be quickly detected by determining the cubic formula representing the right and left boundary contain the obstacle points. The position indicated by "x" in Figure 4 is one that can be travelled as much as possible along the path without collision. If collision exists, the path generation algorithm for collision avoidance is executed to avoid the obstacle.

### B. Path generation using cubic polynomials

To generate a new path using (1), the cubic polynomials, a position and slope of two regions are required. In the collision detection algorithm, an algorithm that avoids the detected collision point generates an avoidance path by setting a path from the control point to the avoidance point to prevent collision with obstacles and setting a path from the avoidance point to the return point over the originally set path. The control and return points are selected from the existing path while the avoidance point is an arbitrary point for avoidance maneuver against obstacles. This point should be capable of generating two homotopic paths which offer the possibility of detouring the collision point, as illustrated in Figure 5. The avoidance point was obtained by using the Voronoi cell-based algorithm. From the avoidance points obtained from the one Voronoi cell algorithm, an avoidance path is generated using the cubic formula thereby verifying whether no collision is expected by using the avoidance path generated by the collision detection algorithm and finally modifying a path that has no collision. If all the avoidance paths are blocked, a new path is searched using VFH (Vector Field Histogram) [15] while driving reversely.

### C. Path generation using Voronoi Cell

Voronoi diagram is for dividing a range having a finite number of obstacles into a number of sub-ranges that each having only one point data while making all sub-regions closest as much as possible one another. Path planning using this is to find an optimized path via the graph search algorithm from the Voronoi edges. Path planning using the Voronoi diagram is not suitable for real time application due to taking a calculation time in an algorithm that searches the Voronoi diagram and the optimal path. Thus it is not appropriate to apply this to point cloud directly which is obtained from real laser scanners. The following algorithm was taken into consideration to employ basic concept of the Voronoi diagram for real-time local path planning within the sensor range. To produce a solution for a collision point, $X_c$, that blocks the path, N numbers of candidates for avoidance points, $X_{avoid}$, can be obtained via the one Voronoi cell algorithm (Figure 6).

An avoidance point is the midpoint between a nearest point ($X_{near}$ closest to a collision point from obstacle point set, $X_{obs}$) and $X_c$. $t_{avoid}$ is a slope of a voronoi edge, $E$. Once obtaining $X_{avoid}$ and $t_{avoid}$, in a voronoi edge, repeat the algorithm after removing sub-elements of $X_{obs}$ which are located outside of the edge until $X_{obs}$ is not null. Since the one Voronoi cell algorithm finds edges in only a single Voronoi cell in contrast with general Voronoi Diagrams, it has an advantage that can be used in real time due to fast calculation speed. Moreover, it searches the maximum clearance against obstacles according to the definition of the Voronoi Diagram so that it provides avoidance path of complex obstacles environment stably. Furthermore, it can increase efficiency of a path by limiting the size of a cell to prevent a long detour due to a long gap between obstacle points.

*One Voronoi Cell Algorithm* ( $X_{obs}, X_c$ )

$n \leftarrow 0$
*do*
  $n \leftarrow n + 1$
  $X_{near}(n) \leftarrow GetNearestPoint(X_{obs}, X_c)$
  $(E(n), X_{avoid}(n), t_{avoid}(n)) \leftarrow GetVoronoiEdge(X_{near}, X_c)$
  $X_{obs} \leftarrow RemovePointSet(X_{obs}, E(n))$
*while* ( $X_{obs} \neq$ Null )
*return* ( $X_{avoid}, t_{avoid}$ )

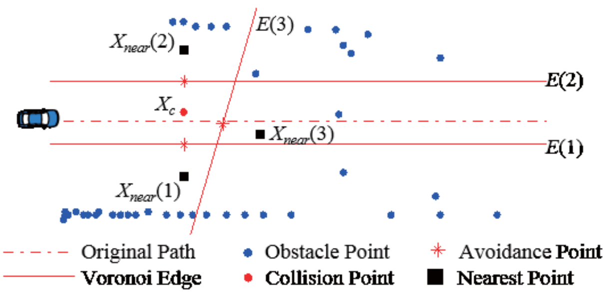

Fig. 6. Path plan using the Voronoi Edge

## IV. EXPERIMENT RESULT FROM COMPETITIONS

The second autonomous self-driving vehicle competition was held on the 20th and 21st of September 2012 at the course in Namyang Research Center of Hyundai Motor Company, Korea. The results of two days main competitions were aggregated as a ratio of 3:7 to determine a rank. The competition venue is shown in Figure 9, which is a course in the Research Center of Hyundai Motor Company, Korea. The competition course is a 3.4km mixed with pave and unpaved roads while the competition mission consisted of nine missions in total such as overtaking slow vehicles, unexpected obstacle avoidance, sensing of traffic lights in the crossing, and parking. The final winner was judged by summing the driving time and success of failure of given nine missions. The missions of the Second Competition were more difficult than those in the First Competition.

For example, sensing of traffic lights in the crossing, sensing of right and left turn, and sensing of passengers in queue required advanced image processing capability while obstacles avoidance and detour required sophisticated path generation and tracking. In addition, unexpected obstacle and overtaking slow vehicles required capabilities of sensing of dynamically moving obstacles and generating an avoidance path dynamically. The last mission was to park automatically at the designated parking area indicated by a number in the road sign. The complex obstacle avoidance mission was to pass the driving road that was blocked with red cones as shown in Figure 10 while the detour mission was to detour the existing path. Figure 7 and Figure 8 show the local path planning results for the complex obstacles and detour missions in the Hyundai self-driving vehicle competition using the proposed algorithm. It was verified that path planning was successfully done

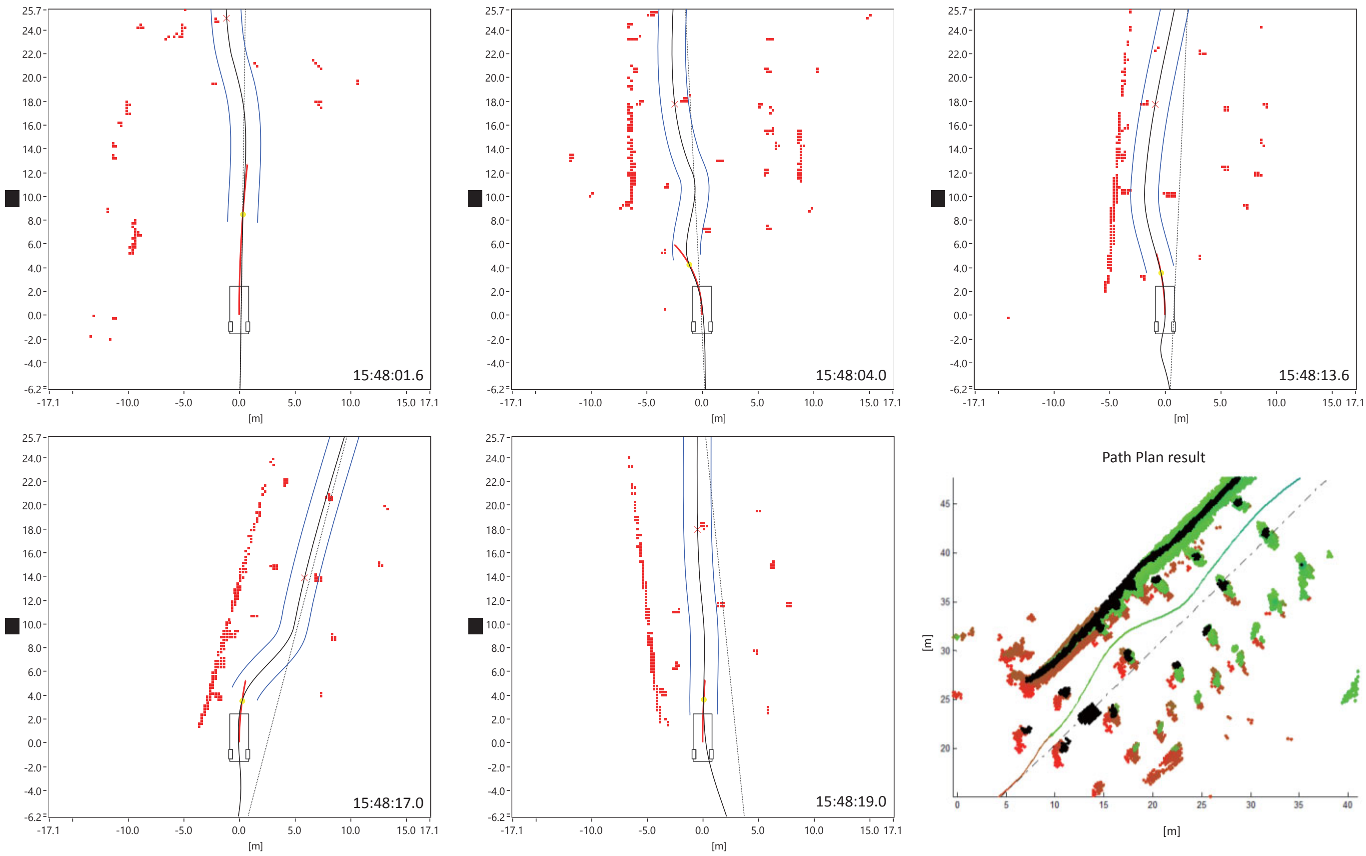


Fig. **7.** Path Planning Result for the Complex Obstacles Mission

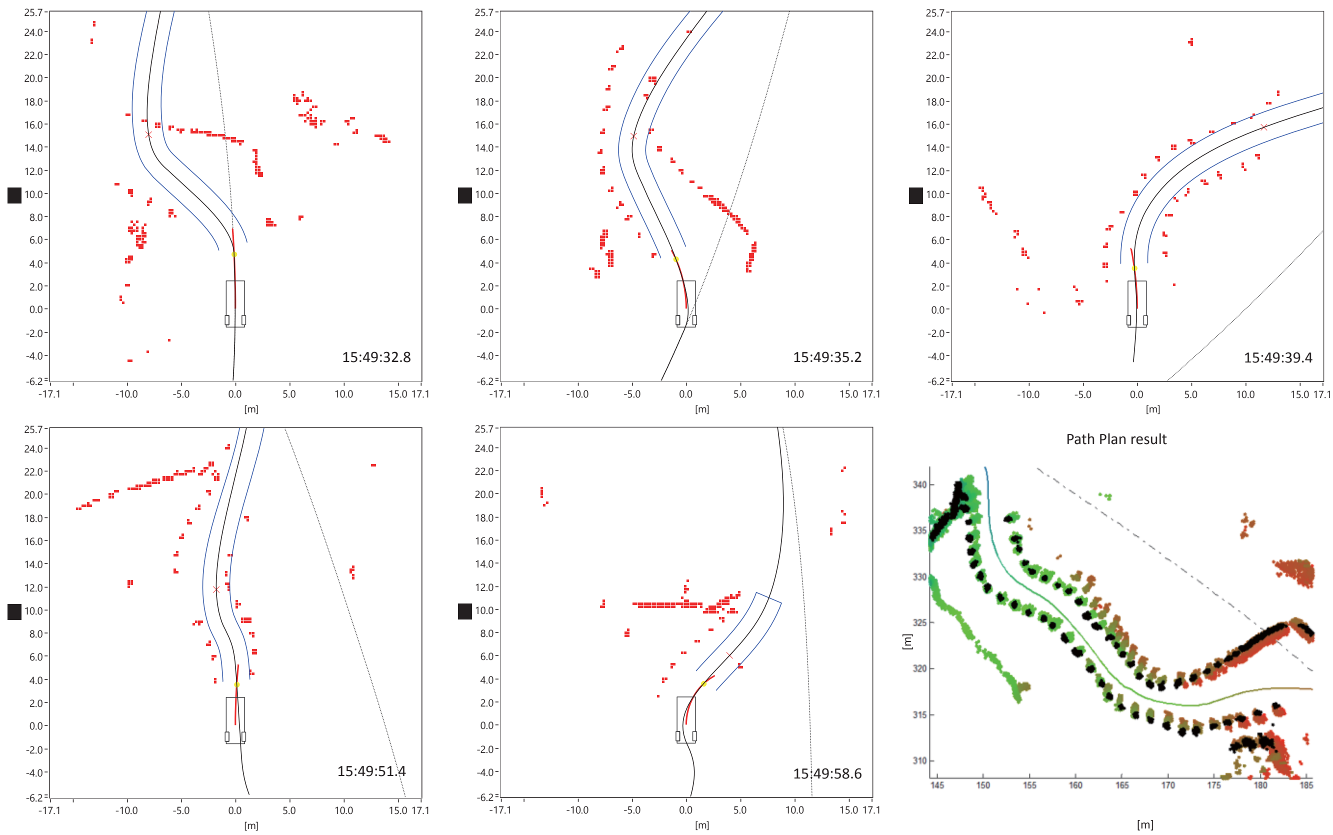


Fig. 8. Path Planning Result for the Detour Mission

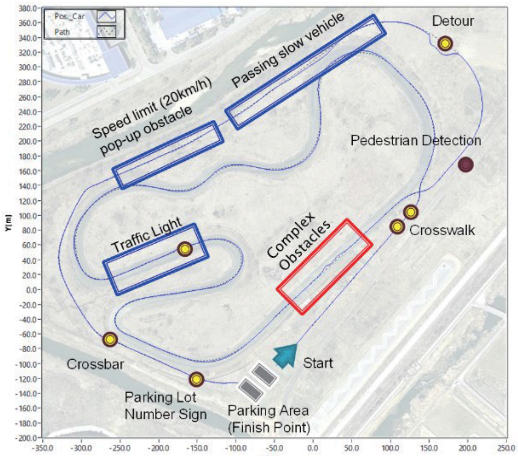

Fig. 7. Hyundai Autonomous Vehicle Competition Course


Fig. 8. Mission of the Second Self-driving vehicle competition (Complex obstacle mission (above), detour mission (below))

## V. Conclusion

This study presented real-time path modification via the proposed cubic formula and successful avoidances under various obstacle environments via the path generation algorithm.

1) This study showed that path management using cubic formula was more efficient and path modification was easier than way points.

2) Continuous detection of collision with obstacles can be possible with path representation using the proposed cubic formula.

3) Path generation under various obstacle environments can be possible using the one Voronoi cell algorithm and VFH as well as real-time use due to reduction of computation.

In the future, our study will develop a self-driving capability as capable as human does thereby sensing surrounding environment more accurately under a number of variables such as weather, road condition, climate, and time zones.

## Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST). (No.20110031920).

## References

[1] J. Wang, S. Schroedl, K. Mezger, R. Ortloff, A. Joos, and T. Passegger, “Lane Keeping Based on Location Technology,” IEEE Transactions on Intelligent Transportation Systems, vol. 6, no. 3, pp. 351–356, September 2005.

[2] Li, TS and Yeh, Ying-Chieh and Wu, Jyun-Da and Hsiao, Ming-Ying and Chen, Chih-Yang, “Multifunctional intelligent autonomous parking controllers for carlike mobile robots”, IEEE Transactions on Industrial Electronics, Vol. 57, No. 5, pp. 1687-1700, 2010.

[3] Kwanghyun Cho, Seibum B. Choi, "Design of an Airbag Deployment Algorithm Based on Pre-Crash Information", IEEE Transaction on Vehicular Technology, Vol. 60, No. 4, pp. 1438-1452, 2011.

[4] Ü. Özgüner, C. Stiller, and K. Redmill, “Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience,” Proc. IEEE, vol. 95, no. 2, pp. 397–412, Feb. 2007.

[5] Thrun, S., et al., "Stanley: The robot that won the DARPA Grand Challenge", Journal of Field Robotics, 23(9), 661–692, 2006..

[6] J. Wit, C. D. Crane, and D. Armstrong, “Autonomous Ground Vehicle Path Tracking,” Journal of Robotic Systems, vol. 21, no. 8, pp. 439–449, 2004.

[7] O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,” The International Journal of Robotics Research, vol. 5, no. 1, pp. 90–98, March 1986.

[8] S. Koenig and M. Likhachev, “Fast Replanning for Navigation in Unknown Terrain,” IEEE Transactions on Robotics, vol. 21, no. 3, pp. 354–363, June 2005.

[9] N. S. . V. Rao, N. Stoltzfus, and S. S. Iyengar, “A Retraction Method forLearned Navigation in Unknown Terrains for a Circular Robot,” IEEE Transactions on Robotics and Automation, Vol. 7, No. 5, pp. 699–707, October 1991.

[10] L. E. Kavraki, P. ˇSvestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces,” IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566–580, August 1996.

[11] H. M. Choset, Principles of Robot Motion: Theory, Algorithms, and Implementation. Cambridge, MA: MIT Press, 2005.

[12] L. Zhang, Y. J. Kim, and D. Manocha, “A Hybrid Approach for Complete Motion Planning,” in IROS, October 2007, pp. 7–14.

[13] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path planning for autonomous vehicles in unknown semistructured environments,” Int. J. Robot. Res., vol. 29, no. 5, pp. 485–501, Apr. 2010.

[14] W. L. Nelson, “Continuous steering-function control of robot carts,” IEEE Transactions on Industrial Electronics, Vol. 36, No. 3, pp. 330–337, 1989.

[15] Borenstein, Johann and Koren, Yoram, “The vector field histogram-fast obstacle avoidance for mobile robots”, IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, pp. 278-288, 1991