Environment-Detection-and-Mapping Algorithm for Autonomous Driving in Rural or Off-Road Environment

Jaewoong Choi, Junyoung Lee, Dongwook Kim, Giacomo Soprani, Pietro Cerri, *Member, IEEE*, Alberto Broggi, *Senior Member, IEEE*, and Kyongsu Yi

Abstract—This paper presents an environment-detection-and-mapping algorithm for autonomous driving that is provided in real time and for both rural and off-road environments. Environment-detection-and-mapping algorithms have been designed to consist of two parts: 1) lane, pedestrian-crossing, and speed-bump detection algorithms using cameras and 2) obstacle detection algorithm using LIDARs. The lane detection algorithm returns lane positions using one camera and the vision module "VisLab Embedded Lane Detector (VELD)," and the pedestrian-crossing and speed-bump detection algorithms return the position of pedestrian crossings and speed bumps. The obstacle detection algorithm organizes data from LIDARs and generates a local obstacle position map. The designed algorithms have been implemented on a passenger car using six LIDARs, three cameras, and real-time devices, including personal computers (PCs). Vehicle tests have been conducted, and test results have shown that the vehicle can reach the desired goal with the proposed algorithm.

Index Terms—Autonomous driving, lane detection, obstacle detection, pedestrian-crossing detection, speed-bump detection.

I. INTRODUCTION

AUTONOMOUSLY driven cars have been researched for quite some time. An autonomous ground vehicle has a number of benefits to society, including prevention of road accidents due to human error, optimal fuel usage, and convenience.

However, when trying to make a vehicle autonomously travel to a predefined destination, several challenges need to be overcome. The first need is to know where the vehicle is to detect where the predefined destination is. The second need is to detect surrounding environments to avoid a collision. The third need is to detect signs on the road, such as lanes, crosswalks, and speed bumps, particularly in a rural environment.

In recent years, the Defensive Advanced Research Projects Agency (DARPA) has taken the lead in encouraging research in this area [1], [16], [19].

Vision fields increase their importance each day. These systems met new life from the 1980s/1990s, where their use is associated with safety systems, control, and automation. Some of the first nations that have started this development were Germany (Ernst Dickmanns and VaMP [24]), Italy (ARGO Project [25] and VIAC [28]), the European Union (EUREKA Prometheus Project [26]), the U.S., and Japan, among others. Vision introduces other kinds of sensing features in typical radars or LIDARs [27].

Most obstacle- and ground-detection algorithms use a global mapping algorithm using Global Positioning Systems (GPSs)/inertial measurement units, LIDARs, and cameras. These algorithms detect an obstacle's position in vehicle coordinates, transfers them to global coordinates, and uses the global map to determine the path [9], [10]. Several algorithms, particularly those that consider moving obstacles and pedestrians, use the local window to determine the obstacle map [8]. Most of these algorithms generate a world model map separated by cells and use the concept of confidence [3], [12], [13].

In most research, the data density of LIDARs, three-axis transformation, three-axis rotation angle error from GPS/Inertial Navigation Systems (INSs), and the computational burden come out as a problem, particularly for high-speed driving or off-road or bump driving. To overcome the data-density problem, several approaches, such as plane fitting, combining stereo cameras and LIDARs, differential depth, and slope, have been adjusted [2], [6], [7], [20]. To reduce pose error in 3-D point cloud acquisition, several studies use simultaneous localization and mapping, iterative closest points, and scan matching [4], [11], [14], [17], [18]. These algorithms use the overlap of the detecting area and the mismatch error of the overlap. However, it is not suitable for off-road driving, which does not have enough corresponding points. Therefore, several algorithms try to consider and overcome the errors from GPS/INS with probabilistic approaches [5]. Stanford University, Stanford, CA, which won DARPA 2005, uses the probabilistic terrain analysis algorithm, which is based on the probabilistic error model (a Gaussian) for terrain labeling [21]. It successfully reduces the terrain classification error.

J. Choi, J. Lee, D. Kim, and K. Yi are with the Vehicle Dynamics and Control Laboratory, School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Korea (e-mail: loar00@snu.ac.kr; kyi@snu.ac.kr).

G. Soprani, P. Cerri, and A. Broggi are with the Artificial Vision and Intelligent System Laboratory (VisLab), Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Parma, 43124 Parma, Italy (e-mail: cerri@ce.unipr.it; broggi@ce.unipr.it).

Fig. 1.   SNUCLE.

Fig. 2.   Sensor system.

For the LIDAR part, the obstacle detection algorithm is designed with risk estimation and a model-based filter algorithm. The risk estimation algorithm consists of two algorithms: 1) the height-difference-based algorithm and 2) the classification algorithm [15], [22]. These two algorithms are combined based on "risk."

The most important features of the proposed perception algorithm based on LIDAR detection are robustness and expandability. The proposed perception algorithm is designed to have an open structure and to integrate each of the perception algorithms based on "risk" to overcome the tradeoff between computational burden and accuracy of detection. The proposed structure in this paper is designed to control the tradeoff between computational burden and accuracy of detection by the determination of several tunable parameters and minor changes in structure. A new algorithm based on the feature of course condition can be applied to the structure based on the risk.

The proposed algorithms are implemented on a test vehicle, which has six LIDARs, three cameras, and Real-Time devices, including personal computers (PCs), that are tested in real time.

II. VEHICLE SYSTEM CONFIGURATION

A Hyundai Motors Grandeur TG was used as the base platform for the autonomous vehicle SNUCLE. The Grandeur TG contains a typical six-cylinder gasoline engine and front-wheel-drive system. To install various sensors and protect them from accidents, SNUCLE has been outfitted with a front bumper guard and a roof rack, as shown in Fig. 1.

A custom-made device enables direct electronic actuation of a brake with a wire and motor. A throttle actuator is operated by electrical signals from a longitudinal controller. A dc motor attached to the steering column provides the desired steering angle determined by a lateral controller.

Two SICK LIDARs and one IBEO LIDAR are installed at the front bumper guard of the vehicle, and three SICK LIDARs are installed at the custom-made roof rack. Three cameras are installed behind the windscreen of the vehicle. Two cameras are used to detect bumps and pedestrian crossings in front of the vehicle. One color camera is used to detect lanes. This sensor system is shown in Fig. 2.

The computing system of the SNUCLE is located at the back seat. The computing system consists of three networked computers installed in the back seat.
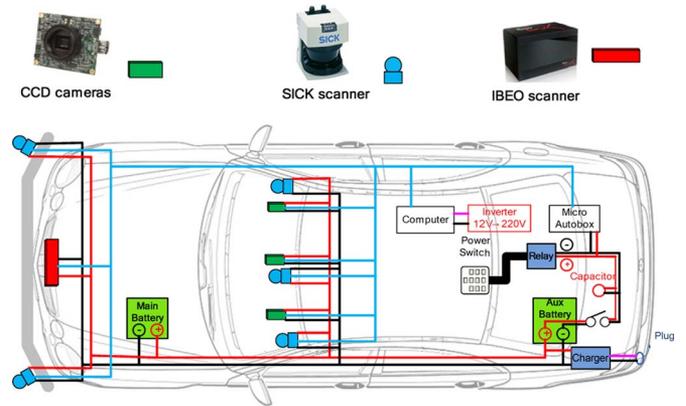
Information on the position and motion of the vehicle is reported by an Oxford Technical Solutions RT3002. The RT3002 outputs its real-time measurements over a controller-area-network bus interface. The RT3002 ensures that the vehicle is aware of its position with a best-case accuracy of less than 0.02 m.

III. VISION DETECTION

The main objective of the vision system is the detection of lane and road markings (pedestrian crossings and speed bumps are considered), which are not easily detectable by LIDARs.

Three cameras are used, i.e., a grayscale camera for pedestrian-crossing detection, a color camera for bump detection, and a color camera for lane detection. Bump and lane detection uses different cameras because the lane detection algorithm runs on an embedded module, whereas bump detection runs on a PC.

The pedestrian-crossing detection algorithm was also tested in usual urban scenarios, with a relaxed set of thresholds (which is, however, different for Italian and Korean environments because of different rules of the road), and it reaches appreciable results: The detection rate is of about 85%, with no false positives. The lane detection results have already been presented in a previous paper [23].

A. Lane Detection

Lane detection runs on a digital-signal-processing-based module called the VisLab Embedded Lane Detector (VELD). Images framed by the camera are processed by the VELD module, and the lane position is returned on a network socket. The lane position is then filtered to reduce noise, assuming that it does not rapidly change in a normal paved road, except in front of intersections.

VELD: The VELD module is based on the elaboration of dark–light–dark (DLD) transitions on an inverse perspective mapping (IPM) image [23]. The IPM transformation allows us to remove the perspective effect from the acquired image, remapping it into a new 2-D domain, in which the information content is homogeneously distributed among all pixels.

The search for lane markings is performed on the IPM image by detecting the DLD transitions and then clustering them
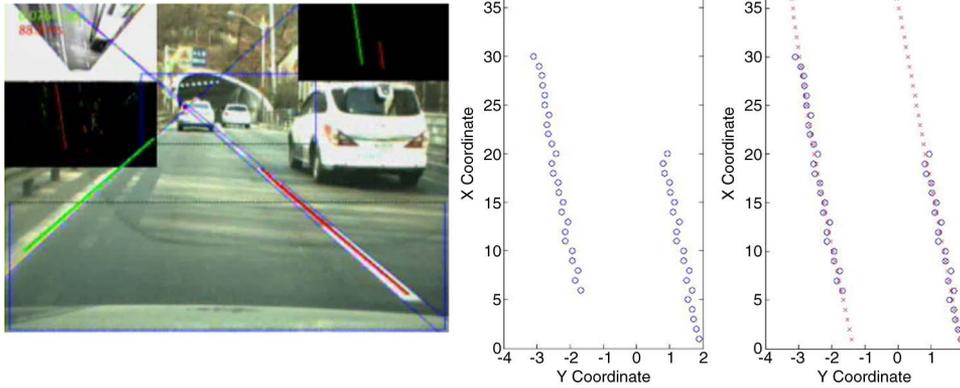
Fig. 3.   Lane data filtering.

according to proximity. The groups of points are approximated using piecewise linear functions.

The lane-marking extraction procedures for dashed and solid lines are slightly different; both of them are built in three phases, i.e., generation, expansion, and validation. The generation phase is different for solid and dashed lines (as the detected clusters are different) but exploits area-based tracking in both cases. Expansion and validation are the same for the two types of lines; the expansion stage is used to connect detected segments to each lane marking, provided that they are compatible with their characteristics.

The validation step is performed by assigning to each lane a score that depends on its length and distance from the ego vehicle.

Lane Data Filtering: The data from the vision module are processed by a filter to minimize the effect of detection inaccuracy. The lane data measured at time k slightly changes according to the movement of the vehicle. If we consider change with sampling time T, then the dynamic model of the road is given by

$$\begin{bmatrix} x_{k,\text{predict}} \\ y_{k,\text{predict}} \end{bmatrix} = \begin{bmatrix} \cos(-\Delta\psi_k) & -\sin(-\Delta\psi_k) \\ \sin(-\Delta\psi_k) & \cos(-\Delta\psi_k) \end{bmatrix} \begin{bmatrix} x_{k-1} - \Delta s_x \\ y_{k-1} - \Delta s_y \end{bmatrix} \quad (1)$$

where

Δs_x longitudinal distance during sampling time T;
Δs_y lateral distance during sampling time T;
$\Delta\psi_k$ rotation during sampling time T.

x is the position of lane in a heading direction represented in a moving coordinate frame attached to the vehicle. In addition, y is the position of lane in a lateral direction represented in a moving coordinate frame attached to the vehicle.

The longitudinal distance during sampling time T is simplified as a product of longitudinal velocity and sampling time T. The rotation during sampling time T is simplified as a product of yaw rate and sampling time T. Then, (1) can be rewritten as follows:

$$\begin{bmatrix} x_{k,\text{predict}} \\ y_{k,\text{predict}} \end{bmatrix} = \begin{bmatrix} \cos(-\gamma_k \cdot T) & -\sin(-\gamma_k \cdot T) \\ \sin(-\gamma_k \cdot T) & \cos(-\gamma_k \cdot T) \end{bmatrix} \begin{bmatrix} x_{k-1} - V_x \cdot T \\ y_{k-1} - V_y \cdot T \end{bmatrix}. \quad (2)$$

The measurement of the lane is updated by the weighted gain determined by the accuracy of the measurement. The process for the lane data is briefly shown in Fig. 3.

B. Pedestrian-Crossing Detection

The method used most to recognize pedestrian crossing is to search its characteristics, such as shape, regularity, or color.

In this case, there is a moving grayscale camera mounted at a height of only 1.22 m. The first step of the algorithm is again to build a top-view image through the use of IPM transformation.

The next step is the processing of the IPM image with a vertical Sobel filter to obtain a new image with only vertical edges.

The third step is the analysis of the edge image to find lines that are almost vertical and longer than an adjustable threshold. This processing is performed for each line that is black or white, which corresponds to the color transitions dark–bright or bright–dark, respectively. All lines are labeled, and their positions are saved.

The following step is label joining, with one white and one black, in a stripe. Specific checks are performed when trying to group together two vertical labels: The first check is on the distance between the extremes, and the second check is on the difference between the orientation of the two labels. Both have to be smaller than specific thresholds.

Afterward, all these blocks are checked and grouped according to rules on distance. Fig. 4 shows the steps of the algorithm.

An additional feature that is aimed at locating missed white dashes is also included. This step is performed by checking if, between two white blocks, there is enough space for a new stripe and if the average intensity where the stripe is missing is similar to the two closer stripes; moreover, the orientation of the two blocks are checked for similarity.

C. Speed-Bump Detection

Detecting a speed bump is quite difficult: The height of a bump is about 0.05 m, and therefore, LIDAR or stereo vision 3-D reconstruction can fail. Moreover, bumps are often stained or scratched, and finally, a number of objects with the same
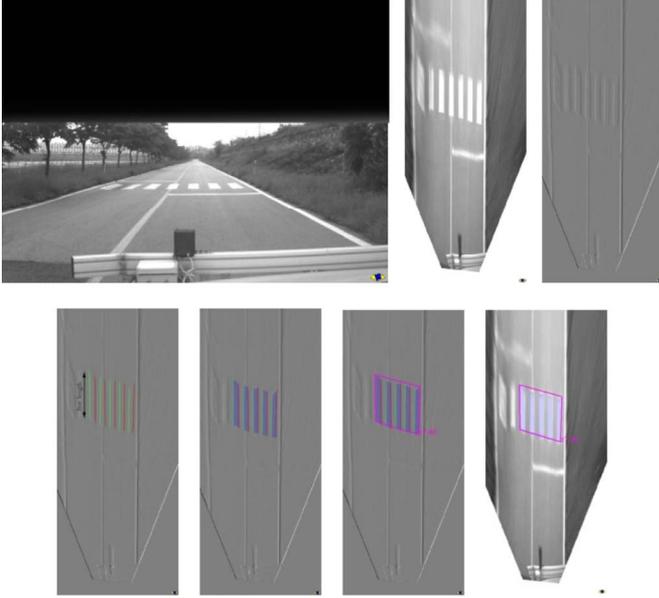
Fig. 4. Full process of the pedestrian-detection algorithm. (First row) Original image, IPM image, and edge detection on the IPM image. (Second row) Results of line detection, stripe detection, and pedestrian-crossing detection, and the final results on the IPM image.

Fig. 5. Output from the speed-bump detection algorithm.

color are present in the road environment. To mitigate the last problem, we decided to analyze only the lower part of the image, i.e., the road. Even if the bumps' height is small, using IPM transformation, their shape gets distorted and stretched.

Speed bumps are formed by diagonal yellow and white stripes. Therefore, a new binarized image with only yellow information is built, considering the ratio between the RGB channels: High values of red and green channels and low values of blue are searched for. A following erosion improves the quality of the image, removing spot noise. Yellow blocks are then built by grouping neighboring yellow pixels.

The next step is to divide the image in horizontal stripes. The maximum gap between each block, the distance between the center of all the blocks, and the nearest blocks on the right and on the left are also computed. Considering the number of detected blocks and the distances computed so far, it is possible to detect speed bumps. The final output is shown in Fig. 5.

IV. LIDAR DETECTION

The purpose of LIDAR detection is to extract positions that cannot or should not be traversed by the autonomous vehicle. Five SICK LIDARs and one IBEO LIDAR are used as the primary obstacle detection sensor for autonomous driving.

The IBEO LIDAR provides a pulse classification algorithm that can be helpful to generate the obstacle map. However, the SICK LIDAR does not contain a ground detection algorithm; therefore, the classification algorithm is designed and implemented on a PC.

The height-difference-based algorithm is applied to detected point clouds, and the label from classification is used to estimate the risk of the points. In specific situations, some assumptions can be added and can help in filtering out misrecognized obstacles. Assuming flat-ground condition, an obstacle can be determined by the z value of the $x-y-z$ pulses. This kind of assumption reduces the possibility of errors, extends the detect length, and fills the unknown area. Therefore, the vehicle is able to travel to the destination faster.

The point cloud, which is labeled by the risk value, is used for obstacle map generation. A model-based filter is designed to handle LIDAR noise. From the risk map of the last time and vehicle movement, the risk map of the present time is derived. Then, the risk-labeled point cloud is used for the measurements to derive the present-time resultant risk map. This map is standard for the obstacle map.

Therefore, the LIDAR detection process consists of six stages. In the first step, LIDAR data are transformed to the local point cloud. In the second step, the classification algorithm is adjusted and labeled for the point cloud. In the third step, the label-based and height-difference-based algorithms are used to estimate the risk of each point. In the fourth step, the estimated risk is integrated using the weighted summation method. In the fifth step, the risk-estimated map is used as a measurement to derive the present-time risk map. In the final step, the obstacle map is derived by the risk map. This process is shown in Fig. 6.

A. Classification Algorithm

The classification algorithm for SICK is used for ground detection. The algorithm labels each detected point as ground or obstacle. The classification algorithm uses the deviation of height z against lateral position y. If the position of the detected points x and z does not change, although the lateral position y changes, those points can be assumed to be ground [15], [22].

First, with a separate scan plane, the best linear fit $z = my + q$ can be calculated. Parameters m and q are calculated with scan points in a window. The number of points in a window is determined by considering the obstacle and road lateral length. The error of linear fit is expressed in the following:

$$E_z = \sum_{i=1}^{N} (z_i - (m_z y_i + q_z))^2 \qquad (4)$$

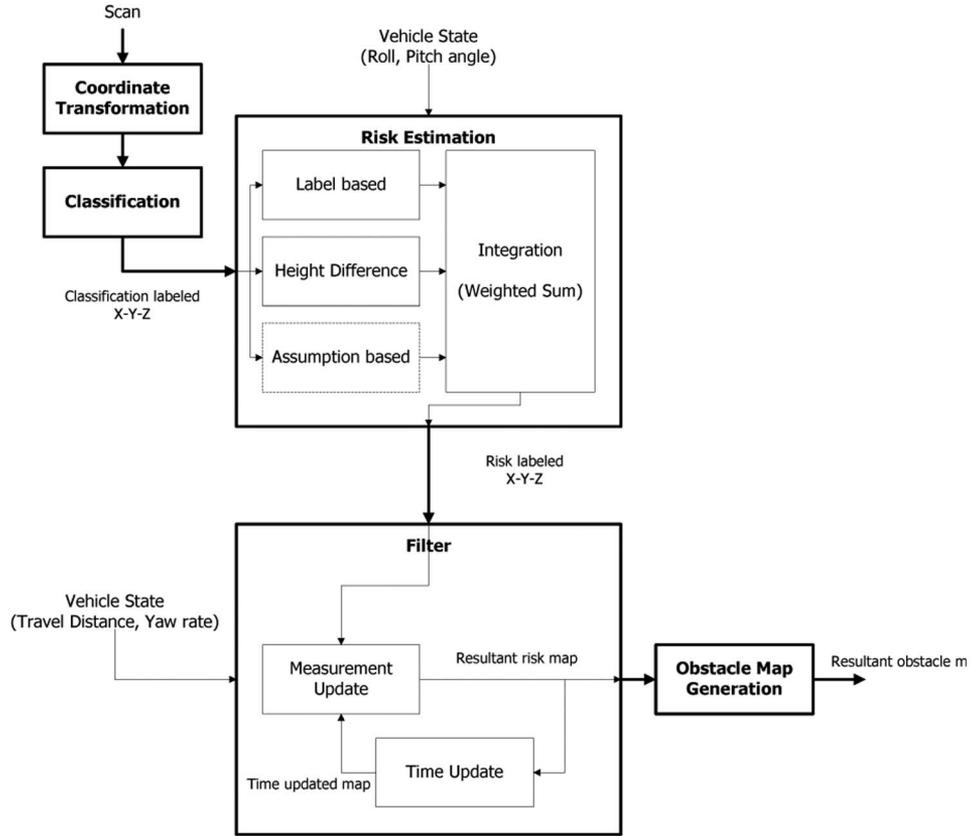$$E_x = \sum_{i=1}^{N} (x_i - (m_x y_i + q_x))^2. \qquad (5)$$

Fig. 6.   Environment map generation process.
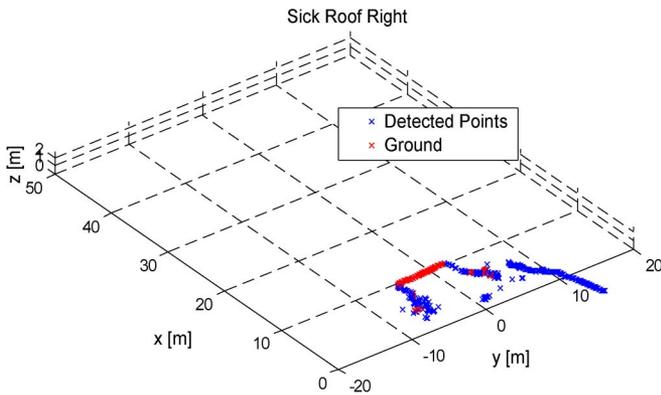

Fig. 7.   Classification results of SICK on right in roof.

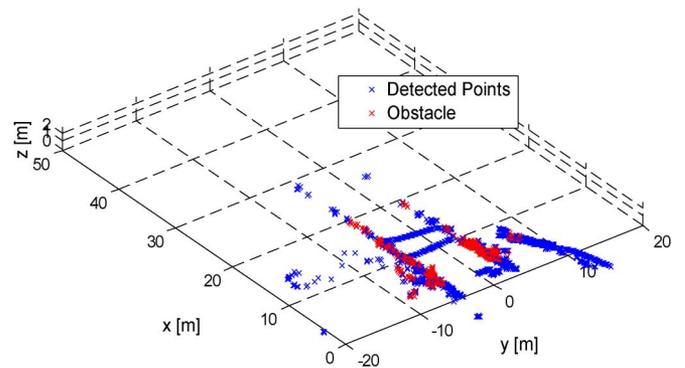Fig. 8.   Detected obstacle points.

The error value is one of the standards for ground detection.

The point belongs to ground if two conditions hold.

1) The linear fit has a small error: $E_z < T_{Ez}$, $E_x < T_{Ex}$.
2) The linear fit has a small slope: $m_z < T_{mz}$, $m_x < T_{mx}$.

Conversely, if the linear fit error or the slope exceeds the threshold, the analyzed point is probably obstacle. The classification result is shown in Fig. 7.

B. Height-Difference-Based Algorithm

The height-difference-based algorithm detects high height-difference points. If the distance between two given points is smaller or equal to a specific predefined range and the z position difference exceeds a critical vertical distance, those two points can be assumed to be dangerous. The critical vertical distance is determined as half of the tire radius. Therefore, the specific range can be determined by this vertical distance and the maximum available climbing angle. The result of the height difference algorithm is shown in Fig. 8.

C. Risk Integration

Risk integration should consider LIDAR installation and the reliability of each algorithm, considering environmental features. The integration algorithm returns the weighted sum of the risk value. The final risk value is expressed as follows:

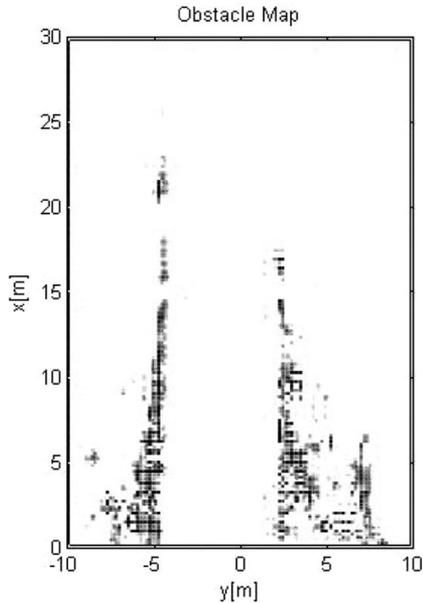$$R_{\mathrm{int}} = W_{gd}R_{gd} + W_{hd}R_{hd} + W_{ab}R_{ab}. \qquad (6)$$

Fig. 9.   Obstacle map.

TABLE I
SITUATIONAL ANALYSIS OF PARAMETERS

Algorithms	Parameters	Analysis
Classification	z-error threshold	Well-organized : Low Off-road : High
	z-slope threshold	No bank : Low High bank : High
	x-error threshold	Well-organized : Low Off-road : High
	x-slope threshold	No bank : Low High bank : High
Height difference	Critical vertical distance	Big Positive Obstacles Only : High Nominal : half of tire radius
Weighted Sum	Weight of Classification	Well-organized : High Off-road : Low
	Weight of Height difference	Well-organized : Low Off-road : High
Risk Map Generation	Obstacle threshold	Well-organized : High Off-road : Low

Well-organized: rural or paved, high speed condition.
Off-road: Not paved or unorganized, low speed condition.

Here, R_{int} is the integrated risk value. R_{gd} is the risk value from the ground detection algorithm. R_{hd} is the risk value from the height-difference-based algorithm. R_{ab} is the risk value from the assumption-based algorithm. W is the weight of each algorithm. It is determined by the reliability of each algorithm and the detected point x, y value.

D. Obstacle Map Generation

The obstacle map is generated using the risk map. If the risk value is bigger than a predefined threshold, it should be an obstacle. The risk map is generated by the risk-labeled point cloud. To handle the noise of data from the LIDAR or environment, such as leaves or sand clouds, it is necessary to compare the present data with the data from the last time. The filter is proposed for this process. The filter integrates the previous risk value with the present risk value and generates the risk map. The resultant obstacle map is shown in Fig. 9. The color of the risk map is determined by the risk value of each grid.

E. Parametric Study

In general, the obstacle detection algorithms have a tradeoff between computation burden and accuracy of detection. The algorithm should be designed to be as simple as it can be; however, the accuracy of the algorithm should be maintained. The risk-based structure, which was developed in this research, has flexibility to apply a new algorithm and can be tuned with parameters based on the road condition. For example, height can be a new standard for risk if the road does not have a high slope and negative obstacles, like a normal rural road.

The parameters of the algorithm can be determined by road conditions. For a rural paved road or organized off-road, which is the normally observed road condition, the risk derived from the classification algorithm is the most dominant. Most of the

obstacles observed under this condition are taller than 1 m. If the road is not in a widely open space, which has a tall building or mountain that can hinder the reception of nearby GPS signals, the time update is not reliable and can be skipped. Under complex road conditions and low speed, the height-difference-based algorithm is more effective than the classification algorithm. These relations can be analyzed, and the parameters can be roughly defined by the road features. These relations are organized in Table I.

V. VEHICLE TEST

The proposed algorithm is evaluated with collected data first, implemented on PCs with the Labview program, and tested in real time. The data are collected in a rural and off-road environment, and the algorithm is tested and tuned for both environments.

A. Off-Road Test

In the off-road test, the cones and bundle of straw are set in the course for obstacles. On the side of the road, reeds and grass can be seen. The roll-and-pitch motion is observed and logged. The vehicle drives about 20–30 km/h in the course without any obstacles in front of the vehicle; however, the high pitch angle error means that the z value of the detected points will be estimated incorrectly. In Fig. 10, the resultant map of the height-difference-based algorithm has some misrecognized obstacles. On the other hand, if the resultant map of the proposed algorithm is adjusted, including the classification algorithm, the ground is rarely misrecognized as an obstacle by this algorithm.

B. Rural Test

In the rural test, lanes, pedestrian crosswalks, and speed bumps must be detected. For such a purpose, computer vision is more useful than LIDAR. This test does not contain intersections and traffic lights.
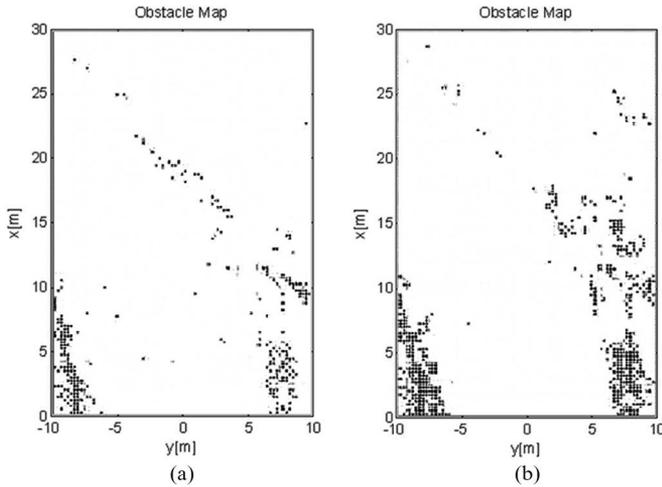
Fig. 10. Obstacle map from the proposed algorithm (high pitch rate). (a) Height-difference-based-algorithm-adjusted map. (b) Proposed-algorithm-adjusted map.
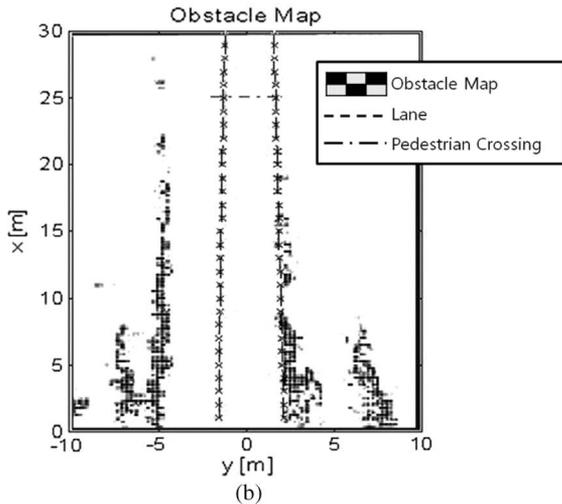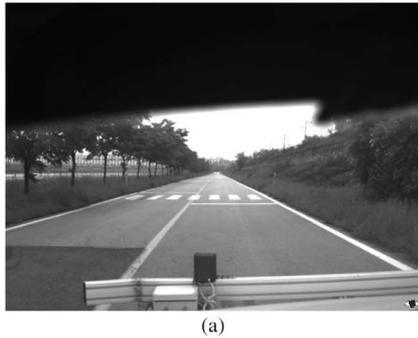

Fig. 11. Environment map. (a) Driving in rural environment. (b) Obstacle map, lane, and pedestrian crossing.

The test environment is shown in Fig. 11(a). The test vehicle drives along the lane, and the vehicle reaches the pedestrian crossing. The obstacle map, driving lane, and pedestrian crossing are shown in Fig. 11(b). The drive lane and obstacle map can be provided to the path-generation algorithm, and the pedestrian crossing is used for velocity control. The autonomous vehicle should drive along the lane and stop in front of the pedestrian crossing.
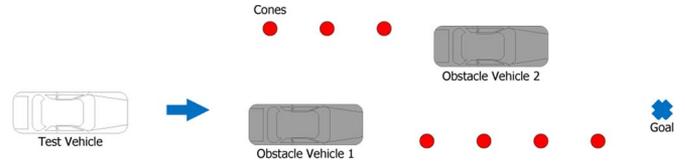

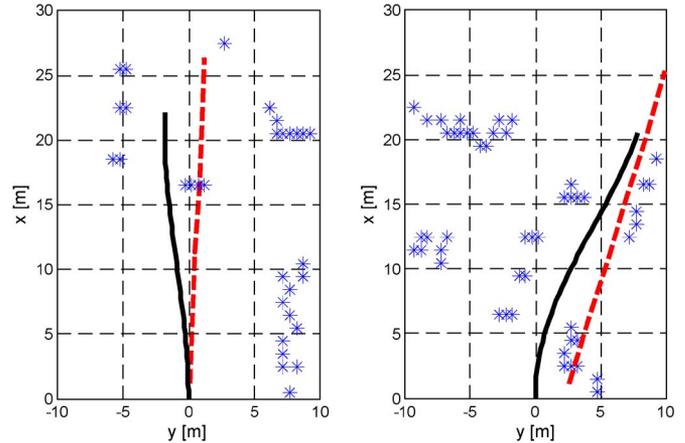Fig. 12. Autonomous driving test environment.

Fig. 13. Generated trajectory and environment map from the path generation algorithm.

C. Autonomous Driving

The test vehicle autonomously drives using the proposed algorithm. The path of the vehicle is determined by the obstacle position and desired goal. The algorithm is designed to use the potential-field-based method. At every time step, the path generator calculates the candidate trajectories using the environment map, vehicle sensor information, and GPS data. The path-generation algorithm selects a path that minimizes cost, consisting of length from the detected obstacles and length between final position of the path and the goal.

The obstacle-detection and path-generation algorithms are implemented on a PC and tested. The first test is applied in a double-lane-change situation shown in Fig. 12. Fig. 13 shows the resultant generated path of the algorithm. The red dashed line is the desired path to the goal position, and the black solid line is the generated path from the path-generation algorithm. The star marker indicates the obstacle points. With the generated environmental detection algorithm, the vehicle successfully avoided obstacles. The path vehicle drives are shown in Fig. 14.

Second, a full test, including off-road and rural tests, has been applied on the test track shown in Fig. 15 at the Hyundai Motor Company. The vehicle drives along the test track while avoiding obstacles. The vehicle drives 3.7 km for 10 min and 53 s. The algorithm is tuned according to the road condition: no negative or hangover obstacles, and nearly flat. Misrecognized points are instantaneously observed and are less than 30 points on the full course. The vehicle detects about 70 points at once; therefore, the probability of misrecognition is lower than about 0.0007%.

Therefore, the misrecognized points do not affect autonomous driving. Table II shows a comparison between the simulation and real tests. The simulation is performed with an ideal sensor model for obstacle detection. The error of the
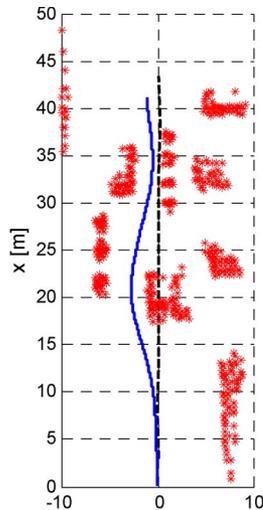
Fig. 14. Driving path of the autonomous vehicle.

Fig. 15. Test track.

TABLE II
COMPARISON BETWEEN SIMULATION AND TEST DATA

Results	Simulation	Test
Driving Time	650 sec	653 sec
Average Velocity	20.77 km/h	20.69 km/h

driving time between simulation and test data is lower than 1%, which means that the perception process shows nearly ideal performance in detecting obstacles.

VI. CONCLUSION

Environment detection algorithms for autonomous driving have been proposed. The algorithms have been designed for static environments of rural and off-road. Environment detection algorithm consists of two parts: 1) vision-based detection and 2) LIDAR-based detection. Vision-based detection has been used for driving-lane detection, pedestrian-crossing detection, and speed-bump detection. The driving lane has been detected by color camera and processed by the VELD module. The lane filter has been designed to estimate the present lane without misrecognition or loss. Pedestrian crossings and speed bumps have been detected by each camera and processed on

a PC. LIDAR detection has been used to detect obstacles. The algorithm has been designed to generate a local obstacle map, instead of a global map, because of difficulties in using accurate vehicle position information. The detected points have been classified, and the risks of the points have been estimated by the classification algorithm and the height-difference-based algorithm. The risk map has been generated by the model-based filter, which uses risk-labeled detected points as measurements. The obstacle map has been generated with the present risk map.

The proposed algorithms have been successfully implemented and tested using a test vehicle. It has been found that the proposed environment-detection algorithm shows good performance in detecting the positions of obstacles, lanes, pedestrian crossing, and speed bumps.

REFERENCES

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffman, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley, the robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, Sep. 2006.

[2] T. Change, T. Hong, S. Legowik, and M. N. Abrams, "Concealment and obstacle detection for autonomous driving," *Int. Assoc. Sci. Technol. Develop.—Robot. Appl.*, pp. 1–15, Oct. 1999.

[3] D. Oskard, T. Hong, and C. Shaffer, "Real-time algorithms and data structures for underwater mapping," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1469–1475, Nov./Dec. 1990.

[4] O. Garcia-Favrot and M. Parent, "Laser scanner based SLAM in real road and traffic environment," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 1–6.

[5] A. Puntambekar, "Terrain modeling and obstacle detection for unmanned autonomous ground robots," Ph.D. dissertation, Oct. 18, 2006.

[6] A. Talukder, R. Manduchi, R. Castano, K. Owens, L. Matthies, A. Castano, and R. Hogg, "Autonomous terrain characterisation and modelling for dynamic control of unmanned vehicles," in *Proc. IEEE/RSJ, Int. Conf. Intell. Robots Syst.*, Oct. 2002, pp. 708–713.

[7] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," Univ. Calif. Santa Cruz, Calif. Inst. Technol., Santa Cruz, CA, 2005.

[8] J. Roberts and P. Corke, "Obstacle detection for a mining vehicle using a 2d LASER," CSIRO Manuf. Sci. Technol., Woodville North, South Australia.

[9] A. Lakhota, S. Golconda, A. Maida, P. Meija, A. Puntambekar, and G. Seetharaman, "Cajunbot: Architecture and algorithms," *J. Field Robot.*, vol. 23, no. 8, pp. 555–578, Jan. 2006.

[10] A. Kelly, *An Intelligent, Predictive Control Approach to the High-Speed Cross-Country Autonomous Navigation Problem.* Pittsburgh, PA: Carnegie Mellon Univ. Press, 1995.

[11] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[12] T. Hong, M. Abrams, T. Chang, and M. Shneier, "An intelligent world model for autonomous off-road driving," *Comput. Vis. Image Understanding J.*, pp. 1–16, 2001.

[13] D. Coombs, K. Murphy, A. Lacaze, and S. Legowik, "Driving autonomously offroad up to 35 km/h," in *Proc. IEEE Intell. Veh. Symp.*, Oct. 2000, pp. 186–191.

[14] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3DIM*, Quebec City, QC, Canada, 2001, pp. 145–152.

[15] J. Weber and L. Matthies, "An obstacle representation for off-road autonomous driving," in *Proc. Intell. Veh. Symp.*, Tokyo, Japan, 1996, pp. 345–350.

[16] A. Petrovskaya and S. Thrun, "Model based vehicle tracking for autonomous driving in rural environments," in *Proc. Robot.—Science Systems IV*, Jun. 2008, pp. 1–8.

[17] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factorized solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, Jul. 2003.

[18] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, pp. 889–916, Sep. 2007.

[19] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y. W. Seo, S. Singh, J. Snider, A. Stentz, W. R. Whittacker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, G. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in rural environments: Boss and the rural challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[20] L. Henriksen and E. Krotov, "Natural terrain hazard detection with a laser rangefinder," in *Proc. IEEE Conf. Robot. Autom.*, 1997, pp. 968–973.

[21] S. Thrun, M. Montemerlo, and A. Aron, "Probabilistic terrain analysis for high-speed desert driving," in *Proc. Robot. Sci. Syst. Conf.*, 2006, pp. 1–7.

[22] F. Maurelli, D. Droeschel, T. Wisspeintner, S. May, and H. Surmann, "A 3D laser scanner system for autonomous vehicle navigation," in *Proc. IEEE Adv. Robot.*, 2009, pp. 1–6.

[23] M. Felisa and P. Zani, "Robust monocular lane detection in rural environment," in *Proc. IEEE Intell. Veh. Symp.*, San Diego, CA, 2010, pp. 591–596.

[24] M. Maurer, R. Behringer, S. Furst, F. Thomanek, and E. D. Dickmanns, "A compact vision system for road vehicle guidance," in *Proc. 13th Int. Conf. Pattern Recognit.*, Aug. 25–29, 1996, vol. 3, pp. 313–317.

[25] A. Broggi, M. Bertozzi, and A. Fascioli, "ARGO and the MilleMiglia in automatico tour," *IEEE Intell. Syst. Appl.*, vol. 14, no. 1, pp. 55–64, Jan./Feb. 1999.

[26] M. Xie, L. Trassoudaine, J. Alizon, M. Thonnat, and J. Gallice, "Active and intelligent sensing of road obstacles: Application to the European Eureka-PROMETHEUS project," in *Proc. 4th Int. Conf. Comput. Vis.*, May 11–14, 1993, pp. 616–623.

[27] A. Broggi, S. Cattani, P. P. Porta, and P. Zani, "A laserscanner-vision fusion system implemented on the TerraMax autonomous vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 111–116.

[28] A. Broggi, L. Bombini, S. Cattani, P. Cerri, and R. I. Fedriga, "Sensing requirements for a 13000 km intercontinental autonomous drive," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2010, pp. 500–505.